Naval Research Laboratory

Washington, DC 20375-5320

NRL/MR/7420--95-7774

XOMNI User and Technical Documentation

DAVID G. BALL
JAMES L. JARVIS
JOHN M. BROZENA
MARY F. PETERS

Marines Physics Branch
Marine Geosciences Division

September 11, 1995

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget. Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 11, 1995	Final

4. TITLE AND SUBTITLE

XOMNI User and Technical Documentation

5. FUNDING NUMBERS

PE - 0603704N
PN - R1987

6. AUTHOR(S)

David G. Ball, James L. Jarvis, John M. Brozena, and Mary F. Peters

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/7420--95-7774

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Stennis Space Center
Bay St. Louis, MS 39529-5060

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

XOMNI is Global Positioning System (GPS) data processing software designed to provide flexible manipulation of all parameters required for GPS solutions. XOMNI was developed based on the OMNI processing system of Dr. G.L. Mader of NOAA. The emphasis of this document is on the differences between XOMNI and OMNI, so this document should be used in conjunction with the OMNI User's Guide (Mader et al, 1991) available from NOAA. A working knowledge of GPS data and calculations is assumed.

XOMNI was developed with one particular type of processing in mind: long base-line kinematic solutions using multiple receivers for cycle slip editing. A set of at least three GPS stations is used at the reference static site and on the Kinematic vehicle being tracked. The redundant stations are used to edit cycle slips.

XOMNI was developed under the HP-UX operating system, a derivative of UNIX. Major parts have been ported to other UNIX systems.

14. SUBJECT TERMS

Software
Kinematic GPS
Carriers phase GPS
Precise positioning

15. NUMBER OF PAGES

107

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18
298-102

CONTENTS

XOMNI TECHNICAL AND USER DOCUMENTATION

INTRODUCTION

XOMNI is Global Positioning System (GPS) data processing software designed to provide flexible manipulation of all parameters required for GPS solutions. XOMNI was developed by Kaman Sciences Corporation for the Naval Research Laboratory (NRL). It is based on the OMNI processing system developed under the direction of Dr. G. L. Mader of the National Geodetic Survey (NGS) in Rockville, Maryland. The emphasis of this document is on the differences between XOMNI and OMNI, so this document should be used in conjunction with the OMNI User's Guide (Mader, 1991) available from NGS. In addition, a working knowledge of GPS data and calculations is assumed.

Although XOMNI includes most of the functionality of the original OMNI, it was developed with one particular type of processing in mind: long baseline kinematic solutions using multiple receivers for cycle slip editing. In this type of processing a set of at least three GPS stations is used both at the reference static site, and on the kinematic vehicle being tracked. The redundant stations are used to edit cycle slips in the data.

XOMNI is being constantly updated and enhanced. It is still a research tool, rather than a production processing system, although it has been used by NRL for production work. XOMNI was developed under the HP-UX operating system, a derivative of UNIX. Major parts of XOMNI have been ported to other UNIX-based operating systems in an ad hoc form, and more formal ports are planned for the future.

This document is divided into the following four sections:

User's Guide: Instructions for using each of the XOMNI main modules, highlighting differences between OMNI and XOMNI.

Kinematic Solution Guide: Step-by-step instructions for generating a kinematic solution using XOMNI.

Differential GPS Solutions: Background information on double-difference phase solutions.

Technical description: Subroutine-level description of the XOMNI software.

Conventions Used in this Document

Filenames

The XOMNI system generates many files and types of files. When file names appear in the text of the document (as opposed to tables) they will be set in **boldface**. Some filenames are based on parameters associated with a particular data set. These parameters will be identified by angle brackets "<>" around a parameter identifier. These parameters include:

<db> database name (four characters), normally a version letter and the day of the year, e.g. **<db>DT.DAT** = "a172DT.DAT".

<doy> Day of year. Actually this is the last three letters of the database name, but it is usually the day of the year, e.g. **MBL<doy>.PLT** = "MBL172.PLT".

An asterisk will be used to identify parts of filenames that are variable, e.g. ***.DAT** might be "nrl2172a.DAT". Note that under UNIX, filenames *are* case sensitive

Modules

Module (main program, subroutine, function) names will be given in all CAPITALS, except when refereing to executable files, i.e. the command entered to run a program. In this case the file name convertion (**boldface**) will be used. For example: MERGE refers to the program module MERGE and **merge320c** refers to an executable file which is entered as a command to run MERGE.

USER'S GUIDE

Again, it should be stressed that this document should be used in conjunction with the OMNI User's Guide. The basic operation of XOMNI is the same as that of OMNI. The main difference is that the menu interface used to run the various modules that make up XOMNI/OMNI is not available in XOMNI. The individual modules are run as individual programs. This allows easy batch processing and an ability to access various versions of the same module simultaneously while enhancements are being tested.

There are two steps to executing each main XOMNI module. First a setup program is run to generate a setup file, then the module itself is run to do the actual processing. The setup programs share some common functionality. Each should be run on a ANSI based terminal. Under HP-UX X Windows an xterm should be used. If a tty type terminal is being used it should be a VT100 type or compatible. The setup programs present the user with a full screen interface. The cursor motion (arrow) keys are used to move between items. The space bar is used to select and item, and the return key is used to terminate a selection. See the OMNI User's Guide for more information. There is a problem entering data into these screens in that in many cases the data being entered is not echoed to the screen. This is really only a problem when entering fields with multiple values, such as dates and times. In this case the user should separate each value with a single comma instead of spaces as specified by OMNI User's Guide.

The command to run the setup program for each XOMNI main module is the first three letters of the name of the module (in lower case) with **set** appended, e.g. the setup program for MERGE is **merset**. The name of the setup files generated by these programs are the full module name (in upper case) with **.INP** appended, e.g. the setup file for MERGE is **MERSET.INP**.

MERGE

MERGE is used to combine data from several stations along with broadcast ephemeris data and optionally precise orbit data into a single database. Additionally various corrections are applied to the data and plot files are generated that can be used to determine data quality, satellite coverage and other information.

Before running **merge**, **merset** is run to generate the setup file, **MERGE.INP**. All parameters for XOMNI **merge** are the same as for the OMNI version. There are several differences in its operation however. These are detailed here.

KINSLV file.

When one station's solution status is set to MBL, or mobile, a file named **KINSLV.OUT** is generated. This is an ASCII file containing one line for each database record generated. The lines contain time, position, velocity, and data quality information. Note that only one station may be designated as MBL at a time. This file can be used as a quick kinematic solution. The positions are from a differential pseudo-range solution and can have fairly large errors, while the velocity is from doppler phase data and is in general more accurate. The following is an example line for a **KINSLV.OUT** file:

```
203 18:23:45.00  245354.234  2342323.343  123213.324  156.345  -89.345  102.343  1.67  1.23
```

The fields are: Day/Time, XYZ position, XYZ velocity, dilution of precision, and dilution of velocity. The last two fields indicate the error contribution due to geometry of the stations and satellites for this epoch. In general, multiplying the error in the raw input data by these numbers will give you the final error of the solution.

Doppler data.

XOMNI MERGE generates an additional database file called **<db>AX.DAT** which contains the doppler derived velocity data for the station designated as MBL. This data is used by NAV22 to do kinematic mode editing and to skip over cycle slips that cannot be fixed.

Handling of P-Code and C/A-Code

OMNI did not differentiate between P-Code and C/A-Code pseudo-range data. XOMNI MERGE comes in three version, currently **merge320c**, **merge320cp**, and **merge320p**. The "c" version only uses C/A-Code, the "cp" version uses P-Code if available and C/A-Code otherwise, and the "p" version only uses P-Code. For C/A-Code only receivers the "c" version should be used. For P-Code receivers the "cp" version should be used only if satellite coverage is a problem, otherwise the "p" version should be used as the switching between C/A-Code pseudo-range and P-Code pseudo-range can affect the processing.

GPS22

The operation of GPS22 in XOMNI is very similar to it's operation in OMNI. The main difference is that an extra file is produced by XOMNI GPS22, the **res.dat** file, that is not generated by OMNI GPS22. This file is used by FUDGE to generate edit files.

To generate a valid **res.dat** file the multiple reference satellite capability of GPS22 must be used (unless one satellite covers the entire processing time). If FUDGE is going to be used the reference satellite scenario contained in the **SAVIT** file (which is what was actually used) must be compared with the requested scenario in the **GPS22.INP** file. If they differ (which means that one or more the satellites was not available at the requested time) GPS22 must be re-run with a different scenario. Other than checking the reference scenario no other action on the part of the user is required to generate the **res.dat** file

The command to run GPS22 is **gps22e**.

NAV22

The following features are available in XOMNI NAV22 and not in OMNI.

Doppler Phase

XOMNI NAV22 makes use of Doppler phase data to generate edits which are not generated by OMNI NAV22. This function is activated by setting the "Fine Fix" (LFIX parameter) to true. The edits will be written to the **NAV22.EDT** file. In addition the **PRES** plot will contain the epoch to epoch changes in the Doppler residual instead of the output phase residuals.

Multiple Reference Satellites

XOMNI NAV22 allows the specification of multiple reference satellites. This feature was only available for GPS22 under OMNI. This feature is currently not supported by the NAV22 setup program–the **NAV22.INP** file must be edited separately. See the appendix for more information on setting a reference satellite scenario.

Bias control files

XOMNI NAV22 reads two additional input files if they exist. The **bias** file specifies times when the bias values for a particular satellite should be reset. This can be used when an uncorrectable cycle slip is detected in one satellite, but there are enough other satellites available to generate a good position. This file is automatically generated by FUDGE.

The **jump** file specifies times when all satellite biases should be recomputed based on the previous epoch position and the Doppler velocity data. This should be used when a jump occurs in the output solution and the cycle slip that causes it can not be isolated. This file must be generated by hand. The format of the file is: day-of-year, hour, minute, and second separated by spaces. A record with a day-of-year of 999 will terminate the reading of the file.

FUDGE

The operation of FUDGE is fairly simple. First, **gps22e** is run with good positions set for the receivers. **Gps22e** generates a **res.dat** file that is then read by FUDGE. FUDGE is run with no arguments and creates files named **n.edt**, **g.edt**, **bias**, and **log**. The **n.edt** file is used to edit a database in preparation for running **nav22**. The **g.edt** can be combined with the **n.edt** file and used to edit a database in preparation for re-running **gps22**. The **bias** file combined with the bias file from the other receiver set can be used when running **nav22**. The **log** file lists the corrections made by FUDGE in a human readable format.

KINEMATIC SOLUTION GUIDE

This guide is meant to provide sufficient information for a user who has some familiarity with GPS data and UNIX to process a set of raw GPS data from the initial reformat step through to a final kinematic solution. In addition to the XOMNI procedures, some recommendations are made as to file naming and directory setup that can simplify the overall process of obtaining the final solution. Because UNIX is a case sensitive operating system and XOMNI uses a mix of lower and upper case letters, the user should pay particular attention to the format of program and file names. Also, XOMNI generates a large number of data and plot files and requires a substantial amount of working disk space. The source files from a four hour session using eight receivers will exceed 100MB. XOMNI will generate an additional 400MB to 500MB of data during the solution process. It is important that the user have adequate disk space available before starting the solution process and monitor available space during the solution. A simplified depiction of the XOMNI solution process using multiple receivers is shown in figure 1.

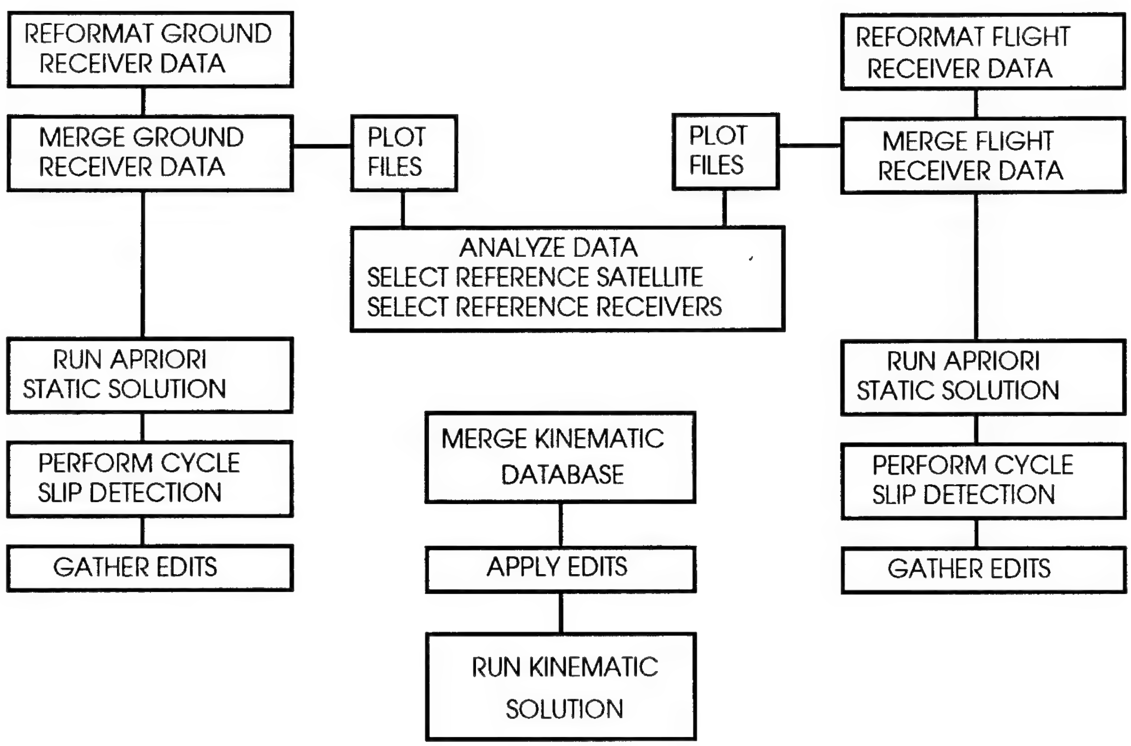

Figure 1 – XOMNI Solution Process

The XOMNI solution process can be simplified by creating a directory structure that separates the fixed and mobile receivers during the static solution and cycle slip detection stage and for carrying out the final kinematic solutions. An additional directory to store reformatted

(RINEX) source data is helpful. A recommended directory setup includes the master directory and four sub-directories as shown below:

jd267

flt gnd kin rinex

A single example of processing data to a final kinematic solution is used throughout the guide which is based on actual data that was collected on September 24, 1993 (Julian Day 267). The example data set, consisting of four receivers from a fixed ground site and four from an aircraft is applicable to the multi-receiver short baseline cycle slip detection scheme used in XOMNI. It is possible to derive a kinematic solution using single fixed and mobile receivers and the differences between single and multiple receiver processing will be discussed in this guide.

Data Reformatting

RINEX stands for the Receiver INdependent EXchange Format, a common data format that allows for the use of data from different sources. In this example, the raw ground receiver data files are named using the Ashtech DOS format and the flight receiver files are named in the NRL data acquisition format, as listed below:

DOS Format		NRL Format	
bgnd1a93.267	**egnd1a93.267**	**a93267.100225h**	**a93267.100225d**
bgnd2a93.267	**egnd2a93.267**	**a93267.100226h**	**a93267.100226d**
bgnd3a93.267	**egnd3a93.267**	**a93267.100227h**	**a93267.100227d**
bgnd4a93.267	**egnd4a93.267**	**a93267.100228h**	**a93267.100228d**

The DOS format data includes b-files (phase data), e-files (orbit) data, and s-files which are a data summary. The s-files are not needed for the solution. The remaining characters in the file name are the receiver name (gnd1), recording session (a), and day and year (93.267). The NRL file format consists of the initial character a, date and time (aYYDDD.HHMMSS) and the final character, d for data, h for header. The NRL data file contains both the phase and ephemeris information. A third file naming convention is used by DOS based data logging systems which record data from multiple receivers to a single PC. These file names have the form;

br10015.267 er10015.267

There will be one pair of these files for each receiver and both are required. It is helpful to rename these files to the standard DOS format (**bgnd1a93.267/egnd1a93.267**) before reformatting.

Raw data is converted to the RINEX format using the program **ashtorin,** which currently can only be used with data from Ashtech receivers. Usage is:

% **ashtorin bgnd1a93.267**

for the DOS format data, or:

% **ashtorin a93267.100325d**

for the NRL format data

ashtorin will respond with a series of prompts which include default values when the required information is available.

file type[]	b, e or s, depending on type of source data - Use the default
site ID[]	Must be entered for NRL format data files (i.e. flt1)
site name[]	Use default for all file types
session[]	Use default for all file types
year[]	Use default for all file types
day[]	Use default for all file types

When all of the prompts have been answered, **ashtorin** will run and create a set of three output files for each input file set. These files include:

gnd1267a.DAT	Phase Data File
gnd1267a.ORB	Orbit Data File
gnd1267a.SUM	Summary File

For the example data set, the rinex directory should now contain the original raw data files plus 24 new files created by **ashtorin**. The raw data files can be deleted if disk space is needed.

Database Creation

As mentioned earlier, processing the data sets from the fixed and mobile receivers is carried out in separate directories because a number of filenames will be duplicated. Carrying out this processing concurrently in separate windows works well and allows for the comparison of the two data sets. Also, processing is simplified if common start and end times are used throughout the solution process. These times can be easily extracted from the .SUM files and combined in a single file using the command:

% **grep TIME *.SUM > jd267.times**

This will create the file **jd267.times** which will contain the start and end times of the eight sets of source data. The latest start time and the earliest end time should then be used for subsequent merge and solution steps. Rounding the start and end times to the nearest inclusive whole minute also makes processing a little easier, but is not required.

Once start and end times have been determined, set up to create the static databases by moving the ground and flight RINEX files to their respective directories (**gnd** and **flt**). If NGS (precise) orbit data is available, it should also be placed in these directories. Set up the monitor with two xterm windows and change to the **gnd** directory in one, and to the **flt** directory in the other.

Databases are built for the ground and flight data with the program **merge320cp**. The program **merset** is used to create a set of input parameters for **merge320cp**. Type **merset** < enter> to display the merge setup screen.

```
                    CURRENT MERGE SETUP                    [mrst-v1.10]
                     DOY   HR    MN   SEC          DOY   HR    MN   SEC
DB NAME:     START:   0     0     0   .00   STOP:   0     0     0   .00

INTERVAL:     .0    ELV LIM    .0      SV IDs:

BC ORB FILE:            ORBIT TYPE:            EPHEM FILE:

                      STATION SUMMARY   (MASTER CLOCK OPTION: 0)
 #   NAME  EDIT  SOLVE  SVCLK     #    NAME  EDIT  SOLVE  SVCLK



                    EDIT          RUN          QUIT
```

Figure 2 – Merge Setup Menu

To make entries in the setup menu, the space bar is used to select an item, necessary values are typed in, and <enter> is used to save the values. Pressing the space bar with the highlight on EDIT will move you to the STATION SUMMARY selection. Pressing the space bar with STATION SUMMARY highlighted displays the station select menu.

```
                        SELECT STATION FILES

     gnd1267a     gnd2267a     gnd3267a     gnd4267a



     <ARROWS> MOVE - <SPACE> SELECTS - <ENTER> COMPLETES
```

Figure 3 – Select Station Menu

Using the arrow keys, move the cursor to the first receiver. Pressing the space bar will open the Initialize Parameters Window.

```
        INITIALIZE STATION PARAMETERS FOR   gnd1267a

      X,Y,Z (m):        918800.116    -5534941.823   3023072.703

      N,E,U (m):              .000            .000          .000

T(C),P(mB),H(%):            15.000         980.000        75.000

L1-L2 OFFSET (m):             .000

        AUTO EDIT:   NO     YES

 SOLUTION STATUS:    OMIT   REF  YES  MBL

    SV CLOCK COR:    NONE   RNG  PHS  BOTH

------------------------EDIT--------------RETURN----------------------------------
```

Figure 4 – Initialize Parameters Menu

If **merset** has not been previously run, the position, offset and meteorological (POM) data is taken from the ***.SUM** file generated by **ashtorin**. Once merge has been run in a directory, a **.POM** file is created which stores these values. In the case of this example, we are doing the initial setup, and the **.SUM** data is displayed. The receiver X, Y and Z positions can be in error by tens of meters, and correct values should be entered if they are available. Typically, these are calculated by carrying out a static survey between the ground site antennas and a geodetic marker with well known coordinates. If a survey has not been possible, the position is normally derived by averaging the pseudo-range results contained in the b-files. No entry is required for the N, E and U values. Use known temperature, pressure and humidity values if they are available or enter the best estimate if actual values were not obtained. The L1-L2 antenna phase center offset is left at .000.

The bottom three lines are used to specify options that must be selected to run merge. AUTO EDIT selects an automatic cycle slip fixing option. It should never be used for data from a mobile receiver. Because XOMNI provides for automatic cycle slip correction at a later stage in processing, AUTO EDIT is normally left on NO for the fixed receivers.

For XOMNI processing, SOLUTION STATUS is normally left on OMIT during the merge of data from the ground receivers. If the position of one of the ground site receivers is well known, SOLUTION STATUS can be set to REF for this receiver, and set to YES for the other

receivers. In this case, merge will generate triple difference solutions for these receivers and record the position in the data base header file. SOLUTION STATUS should always be left off for mobile receivers.

The SV CLOCK COR option specifies whether the satellite clock correction is to be applied to neither the ranges nor the phases (NONE), the ranges only (RNG), the phases only (PHS), or both range and phase (BOTH). In the case of data from the Ashtech receivers used by NRL, this option is set to BOTH.

When all entries have been made for a receiver, the highlight will move to RETURN. Pressing the space bar will permit selection of the next receiver. When entries are completed for all receivers, press the space bar with the highlight on RETURN, press <enter> to exit from the SELECT STATION PARAMETERS prompt, and use the up arrow to move to the DB NAME entry at the top of the setup menu. The data entered in the top portion of the setup screen applies to all receivers included in the data base.

Because merge is normally used to create an initial database, the database is named using the letter **a** and the day of the data. For the example, DB NAME will be **a267**. There are two options for entering the merge start and end times. A left hand window allows use of the actual time from any of the receivers by just entering the number of the receiver, and a right hand window selects manual time entry. Because specific merge times are desired, manual entry is used. The date and time must be entered in the specific format yy,mo,dd, hr,min,sec - an example entry would be **93,9,24,10,3,0**.

INTERVAL is entered in whole seconds and ELV LIM in whole degrees. Typical values would be a one second interval and a 10 degree cutoff. Selecting SV IDs displays the numbers of all satellites in the constellation. The numbers of those satellites that are contained in the orbit data will be highlighted. Individual satellites can be selected and excluded from the database by using the arrow keys to move the cursor to the desired satellite and pressing the space bar. Press <enter> to complete satellite selection.

Selecting BC ORB FILE will replace the STATION SUMMARY window with a window that displays the name of all ***.ORB** files contained in the current directory. Use the arrow keys to move the cursor to the desired file and press <enter>. Only one orbit file may be selected. The orbit type is selected by toggling the ORBIT TYPE entry with the right arrow key. The choices are BROADCAST or PRECISE. Precise orbit data should be used when available. If PRECISE is

selected, EPHEM FILE will display the names of any precise orbit files contained in the current directory. These files must have a **.EPH** suffix.

When all entries have been made, press <esc> to return to the EDIT RUN QUIT prompts. Select RUN with the cursor and press <enter> to save the merge setup. The output from **merset** is the file **MERGE.INP**, which contains the selected merge parameters. An initial **MERGE.OUT** file is also created.

The database merge program is **merge320cp**. Type the program name and press <enter> to start the merge process. The database files created by **merge320cp** include:

a267DT.DAT	Phase and Doppler Data
a267HD.DAT	Database Header Information
a267AX.DAT	Pseudo-range Data
a267OR.DAT	Satellite Orbit Data

Also created by merge320cp is a set of files with ***.PLT** and ***.LIM** extensions. For the example, these will include:

CLK267	Station Clock Data from Range Residuals
ELV267	Satellite Elevation Data
MBL267	Mobile Receiver Position Data if MBL option was selected
PXA267	Raw One-way Phase Residuals
PXB267	Edited One-way Phase Residuals if EDIT option was selected
PXC267	Tracking Coverage
TDF267	Triple Differences used in Solutions

XOMNI plot data can be examined using the programs **xplot** and **splot.** For the plot files generated by **merge320cp** listed above, use the program **xplot.** Usage is % **xplot ELV267** - the extension is not needed. When **xplot** is started, a pop-up menu will be displayed that allows the users to select the satellite elevation data from any or all of the receivers that are included in the database. Selecting **flt2** in the pop-up menu will produce the display shown in figure 5.

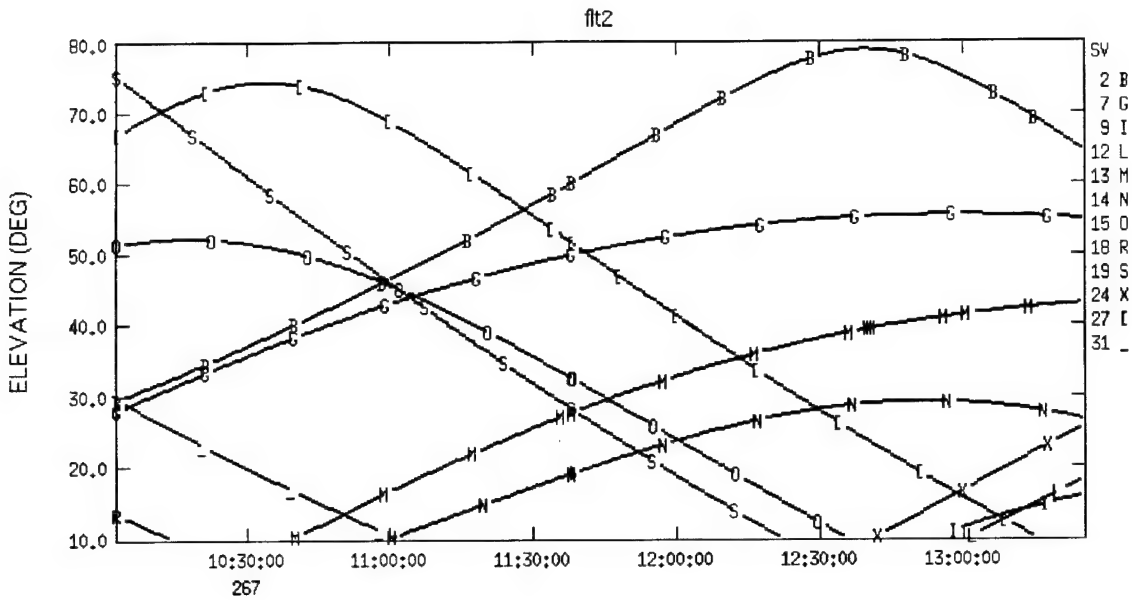

Figure 5 - Satellite Elevation Plot

The ELV plot is used to select a reference satellite for use in subsequent processing. If the data set is short enough, typically less than four hours, a single satellite can be chosen that covers the entire period. For longer data sets, it will be necessary to select additional reference satellites to encompass the full period of the solution. For this example, either satellite 2 (B) or satellite 7 (G) would be a good reference satellite.

The PXC plot should be used to evaluate the data from each group of receivers to determine which should be used as a reference receiver and for use in the static solution. The **xplot** program allows for the simultaneous display of data from all receivers and provides a convenient way to compare the data. The data in the PXC plots should be examined for dropped records and data gaps. The receiver with the best coverage should be selected for use as the reference receiver in the static solution and for later use in the kinematic solution. Figure 6 shows a PXC plot for a single receiver that covers an entire flight.

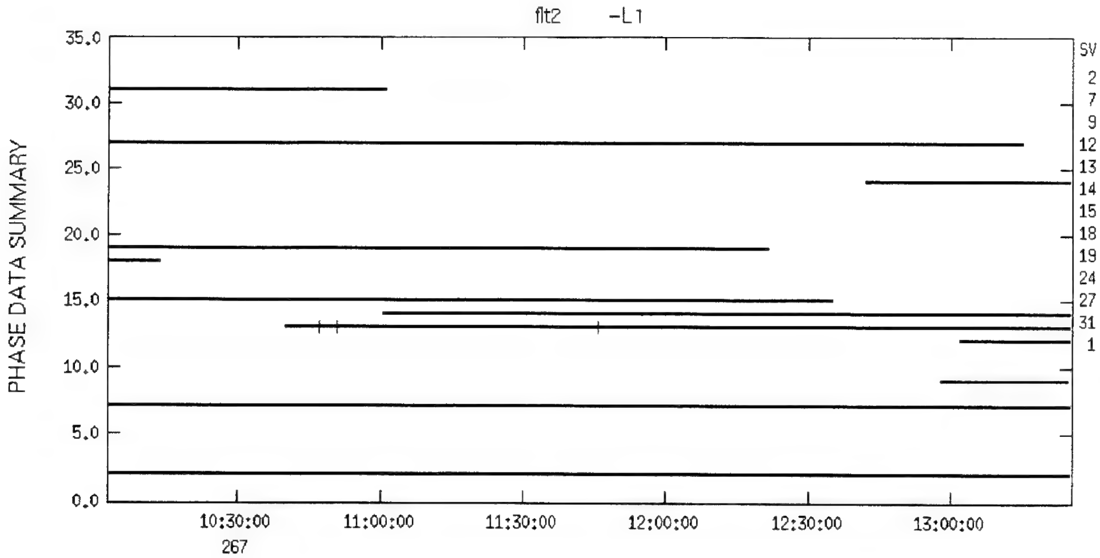

Figure 6 – Phase Data Summary (PXC) Plot

By using the mouse, a portion of the window can be selected and blown up to fill the entire area. Figure 7 shows a section that highlights gaps in the data for satellites 14 and 19.

Examining the **MERGE.SUM** file can also provide clues as to which receiver has the best quality data. **MERGE.SUM** will list all dropped records, and the receiver with the fewest is normally the best choice. If overall data quality is good, there will be no dropped records and the PXC plot must be relied on.

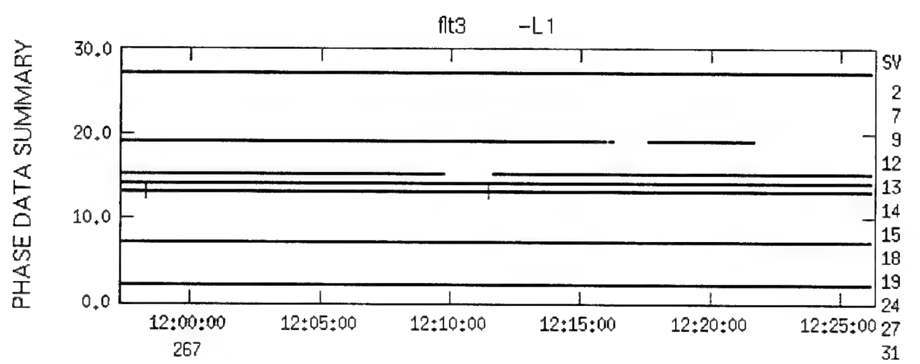

Figure 7 – Expanded PXC Plot

Static Solutions

Once the reference satellite(s) and reference receivers have been determined, the static short baseline solutions can be run. The initial step is to set the clock terms in the **HD.DAT** files to zero

which is done with the command **zero_clocks**. For the mobile receivers, the **HD.DAT** file should be edited and all of the positions should be set equal to the position of the reference receiver.

The program **gpsset** creates the file **GPS22.INP** which contains the input parameters for the static solution program, **gps22e**. Both **gpsset** and **gps22e** should be run from an xterm. Entering % **gpsset** will bring up the **gps22e** setup page as shown in figure 8.

```
              CURRENT GPS22 SETUP                          [gpst-v1.01]

DB NAME:   PROCESSING MODE:              CORRELATIONS:
         DOY:HR:MN:  SEC              DOY:HR:MN:  SEC
START:                          STOP:

FREQUENCY:      TROPO CORR:       ION MODEL:

OMITTED SVs:
ADJUSTED SV ARC ELEMENTS:

 REFERENCE SV #:                      STATION SUMMARY
                   NAME    STAT CLK HGT   NAME     STAT CLK HGT

-------------------- EDIT ----- RUN ---- QUIT------------------
```

Figure 8 – GPS22 Setup Menu

Initially, all entries will be blank, and the steps for entering information are the same as with **merset** - use the cursor keys to move between items, use the space bar to select an item, and use the enter key to save the entered value.

DB NAME will be the name of the database file, in this case, **a267**. For the static editing process, PROCESSING MODE is set to A PRIORIS, and the default NO is used for CORRELATIONS. The times should be the same as used for merge and are entered as DOY HR MIN SEC (267 10 3 0). For FREQUENCY, TROPO CORR and ION MODEL enter the defaults L1, YES and NO. OMITTED SVs and ADJUSTED SV ARC ELEMENTS are left blank. Enter the number of the selected reference satellite in REFERENCE SV. With the cursor on STATION SUMMARY, press the space bar and the first receiver name will be highlighted. Press the space bar again to select the parameters for the receiver, use the right arrow to change

the parameter value, and press the space bar to save the value. The STAT (station) entry is used to specify the reference receiver - REF is used for the reference receiver and SLV (solve) for the others. The CLK term entry is set to MOD (model) and the HGT term is set to FIX for all receivers. When all entries have been made, press < enter> and then <esc> to return to the EDIT prompt. Move the cursor to RUN and press enter to exit **gpsset**.

If more than one reference satellite is needed, an initial **GPS22.INP** file is created using **gpsset**, which will contain the single reference satellite entered in the setup menu. The **GPS22.INP** file must then be manually edited (using vi or another ASCII text editor) to add the additional reference satellites and the start time that each satellite becomes the reference. An additional zero is inserted before and after the lines containing the satellite information. Examples of the two formats are shown in figures 9 and 10 (for more information see Appendix A).

```
        0
  a267
        0
      267    12    33    .0   234    16    10    .0
        1     1     0
    0
        0
       23
        4
  flt1        1     -1    0
  flt2        0     -1    0
  flt3        1     -1    0
  flt4        1     -1    0
```

Figure 9 - Single Reference satellite GPSS22.INP File

```
      0
 a267
      0
    267   12   33   .0  234   16   10   .0
      1    1    0
    0
      0
      0
     23  267   12   33    0
      7  267   15    5    0
     17  267   17   22    0
      0
      4
 flt1       1     -1    0
 flt2       0     -1    0
 flt3       1     -1    0
 flt4       1     -1    0
```

Figure 10 – Multiple Reference Satellite GPS22.INP file

When the **GPS22.INP** file is complete, enter % **gps22e** to run the static solution. This will create a file of residuals, **res.dat**, and a plot file **DDR267.PLT**.

Edits are generated by running the program **fudge**. Typing % **fudge** will create the files **g.edt**, **n.edt**, **n.fdg**, **bias** and **log**. The **n.edt** and **n.fdg** files contain the edits used to correct cycle slips. Because only one of the four receivers in each group is used for the kinematic solutions, the appropriate edits must be extracted from these files for use in editing the kinematic database. For the example, we will assume we have selected the receivers flt2 and gnd3, and we designate flt2 as receiver 1 and gnd3 as receiver 2 for the kinematic solution. The program **gather** is used to extract the necessary edits from the ***.edt** and ***.fdg** files. For the example, the following commands are used:

In the gnd directory

% **gather 3 2 n.fdg > gnd.fdg**

% **gather 3 2 n.edt > gnd.edt**

In the flt directory:

% **gather 2 1 n.fdg > flt.fdg**

% **gather 2 1 n.edt > flt.edt**

The next step is getting set up to run the kinematic solution. A procedure that works well is to collect all of the files required for the kinematic solution in the kin directory before starting the process. These include the *.edt and *.fdg files from the gnd and flt directories, the precise orbit data file, and the RINEX files (DAT,ORB,POM,SUM) for the selected static and mobile receivers. An additional step is creating the combined **bias** file, which can be done from within the kin directory using the command

% **sort +3 ../gnd/bias ../flt/bias | sort -mu > bias**

Kinematic Solution

Creating the **MERGE.INP** file for the kinematic database is done the same way as with the static solutions, but with several differences in the selected parameters. In the STATION SELECT window, the two receivers (flt2, gnd3) will be displayed. The POM information in the top three lines is contained in the .POM files and will already be set, the L1-L2 OFFSET remains zero as before, and AUTO EDIT remains set to NO. For SOLUTION STATUS, REF is selected for the fixed receiver and MBL for the aircraft receiver. SV CLOCK CORR is set to BOTH for both receivers.

In the top portion of the setup screen, use the same database name as was used previously (a267), the same start and end times, interval, and the same satellites. Either receiver can be used for the broadcast orbit file, and the PRECISE option selected if precise orbit data is available. As before, press <esc> to return to the **EDIT RUN QUIT** prompts, select **RUN** with the cursor and press <enter> to save the merge setup.

When run, **merge320cp** will create the **a267** database, the various plot files, and several new files that were not created during the static merge process. These include the **MBL267.PLT**, **MBL267.LIM** and **KINSLV.OUT** files. Run % **zero_clocks** to zero the clock data in the **HD.DAT** file. The **HD.DAT** file already contains the accurate position data for the fixed receiver since it is contained in the .POM file. An accurate starting position will be computed for the mobile receiver later in the solution process.

The next step is to apply the corrections that were generated during the static cycle slip detection process. Because the corrections contained in the ***.fdg** and ***.edt** files act differently on the database, they are applied separately and must be combined into different files.

The commands used are:

% **sort -n +5 flt.fdg gnd.fdg > a.fdg**

and

% **sort -n +5 flt.edt gnd.edt > b1.edt**

The program **cdata** is then run to apply the corrections to edit the database using:

% **cdata a267 b267 a.fdg**

which creates database b267, and

% **cdata b267 c267 b1.edt**

which creates database c267.

An accurate starting position for the mobile receiver is determined by computing the baseline between it and the fixed receiver by running **gps22e** in solution mode during the time period that the mobile receiver was stationary. The pre- and post- flight stationary periods can be determined by examining the **MBL** plot generated by **merge320cp**, using the NORTH or EAST position. Run **gpsset** and set PROCESSING MODE to SOLUTION, and set the times to those of the before flight stationary period. Running **gps22e** creates a file named **SAVIT** which contains the solution results. The **SAVIT** file should be examined to check the quality of the solution, which is indicated by the OVERALL RMS value. This value should be less than .010 for baseline lengths of less than a kilometer. Large rms values usually mean the mobile receiver has begun moving prior to the end time selected for solution, or there is a cycle slip in the data used for the solution. This can be determined by examining the DDR plot generated by **gps22e**. A DDR plot for the case where the static solution was continued past the point where the aircraft started moving is shown in figure 11.

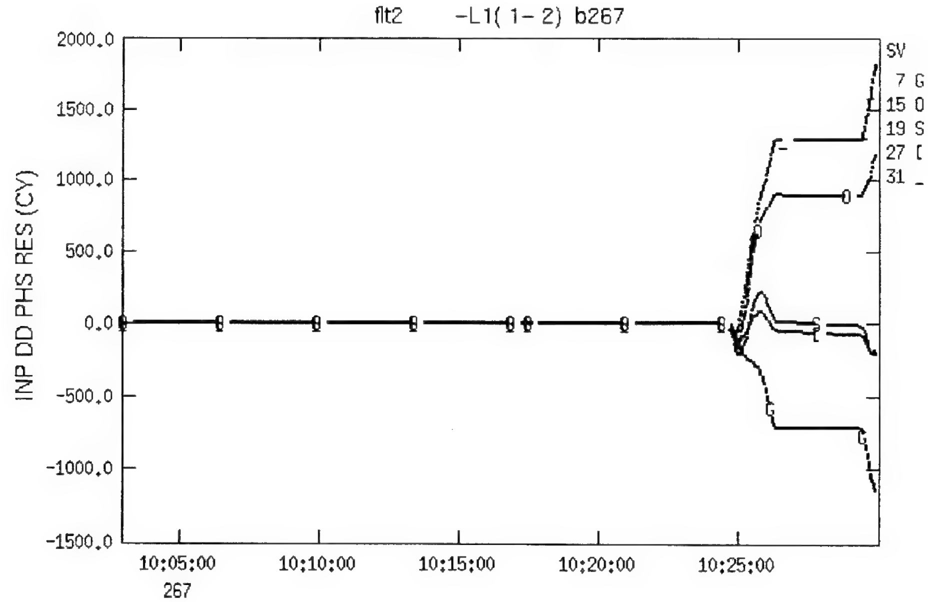

Figure 11 – DDR Plot Indicating Receiver Movement

As can be seen, the residuals diverge rapidly when the aircraft begins moving.

Figure 12 is a DDR plot that shows a cycle slip for SV15 at 10:15:00. The cycle slip would result in a poor quality solution for the static baseline.

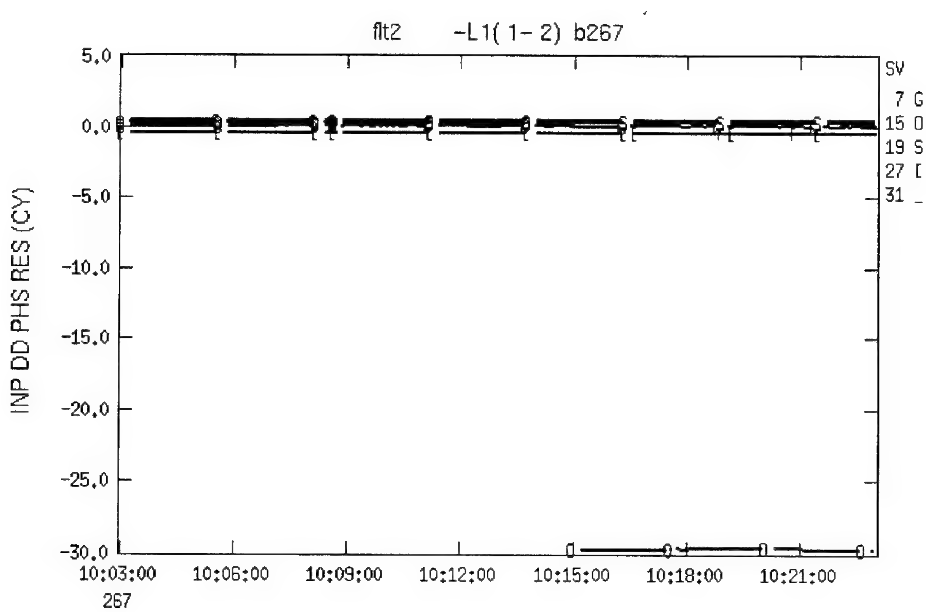

Figure 12 – DDR Plot with Cycle Slip

By examining the DDR plots, a period of time can be selected which will result in a good solution for the static baseline.

Figure 13 is a DDR plot that shows a good static baseline solution but shows the residuals diverging over time. This spread in the residuals for the individual satellites results from not yet having the position of the aircraft receiver accurately entered in the database. When the **SAVIT** file and DDR plot indicate a good solution has been obtained, the aircraft receiver position in the **HD.DAT** file is updated by running % **savit**. Running **savit** also displays the size of the changes made to the X, Y and Z positions in the header file and prompts the user to save or reject the changes. The baseline solution should be run again with this new position, and the DDR plot re-examined. A good baseline solution with an accurate position entered for the mobile receiver in the header file is indicated if the scale range on the plot is 0.1 or less, as shown in figure 14.

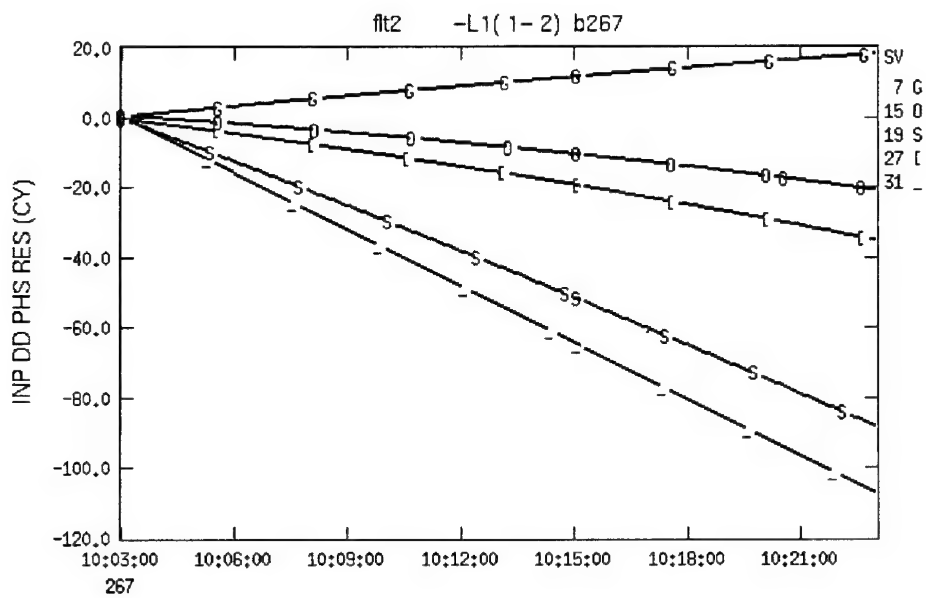

Figure 13 – DDR Plot with Inaccurate Receiver Position

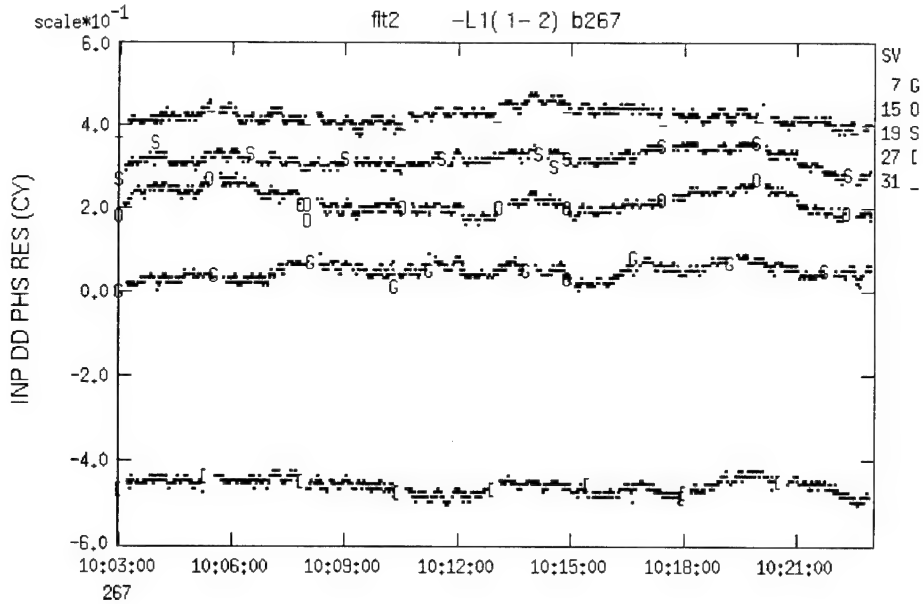

Figure 14 – DDR Plot with Correct Receiver Position

As with the static solutions, a setup program, **navset**, is used to create an input parameters file, **NAV22.INP,** which is used by the kinematic solution program, **nav130**. The NAV22 input menu is shown in figure 15. For the kinematic solution, the DB NAME will be the final edited database (**c267**), PROCESSING MODE will be set to SOLUTION, and PLOT MODE will be set to SOLN ONLY. The first solution will be run with FREQUENCY set to L1 and the times the same as used for the kinematic merge.

```
                         CURRENT NAV22 SETUP              [nvst-v1.00]
DB NAME:               PROCESSING MODE:             PLOT MODE:
                          DOY:HR:MN:  SEC          DOY:HR:MN:  SEC
FREQUENCY:      START:                          STOP:

RNG SOLN:       TROPO CORR:       ANT SWAP:

BIAS VALUES:    FINE FIX:

REFERENCE SV #:                   STATION SUMMARY
                                    NAME      STATUS



------------------ EDIT ---- RUN ---- QUIT------------------
```

Figure 15 – NAV22 Setup Menu

For RNG SOLN, TROPO CORR, ANT SWAP and BIAS VALUES, the default values DIFF, YES, NO, and FLOAT should be used. Set FINE FIX to YES. The same reference satellite should be used as was used for the static editing process. For STATION SUMMARY, the fixed site receiver is entered as REF (reference) and the mobile receiver is entered as SLV (solve).

As with the **GPS22.INP** file, the **NAV22.INP** file must be manually edited if multiple reference satellites are required. Examples of **NAV22.INP** files with single and multiple reference satellites are shown in figures 16 and 17. An important difference to note is that the end times of the satellite reference periods are used in **NAV22.INP**, whereas the start times were used in **GPS22.INP**. Also, the time line is entered as 999 24 0 0 for the last reference satellite in **NAV22.INP**.

```
    0
c267
    1    1
  267   10    03    .0  267    13    45    .0
    1    2     1     0    0     1
   23
    2
flt2            1
gnd2            0
```

Figure 16 – Single Reference Satellite NAV22.INP File

```
      0
c267
      1     1
    267    10    03    .0  267    13    45    .0
      1     2     1     0    0     1
      0
     23   267    12    33    0
      7   267    15     5    0
     17   999    24     0    0
      2
flt2              1
gnd2              0
```

Figure 17 – Multiple Reference Satellite NAV22.INP File

The additional zero is required before the first satellite entry, but not after the last satellite, as in the **GPS22.INP** file.

Computing a kinematic solution is an iterative process that requires running the **nav130** solution program, checking the quality of the solution, determining what corrections need to be made, applying the corrections, and rerunning **nav130**. Once the **NAV22.INP** file has been created, type **nav130** <enter> to run the solution.

If there are no major problems with the data, **nav130** will run to completion and a number of new files will be created. These include the solution files:

NAV22.EDT	Edits generated by nav130
NAV22.OUT	The solution data file
NAV22.SUM	Solution summary file
NAV22.GDS	Currently contains no data

Also created are a number of plot files - the content of some of these files will vary depending on the value of several of the input parameters. These include:

PSOL267	The phase solution data
RDOP267	Solution DOP data

DRNG267 Pseudo-range - Phase solution difference

PRES267 Input or Output phase residuals

There are two primary types of errors that occur in the solution. The first type of error is caused by changes in the geometry of the solution that occur when a satellite is added to or removed from the constellation. In some cases, the result is a jump in the solution at the time the change occurs. These jumps can be detected by running the program **navjump**, which scans the **NAV22.OUT** solution file and produces a listing as shown in Figure 18.

A line will be output for each detectable jump and each time a satellite is added to (start) or removed from (out) the solution. The output from **navjump** is redirected to a file called **jump** with the command:

% **navjump > jump**

As can be seen from the file, not all of the changes to the constellation result in a jump and there are several jumps not associated with a start or out time.

```
10:03:01.000  7 start: 0:  7
10:03:01.000 15 start: 0:  7 15
10:03:01.000 19 start: 0:  7 15 19
10:03:01.000 27 start: 0:  7 15 19 27
10:03:01.000 31 start: 0:  7 15 19 27 31
10:41:16.000          -7.7420
11:01:20.000          -6.1940
11:01:20.000 31 out  : 0:  7 15 19 27 --
11:02:54.000           1.5680
11:08:55.000 14 start: 0:  7 14 15 19 27 --
11:13:57.000           1.8640
12:21:07.000          -2.4120
12:21:07.000 19 out  : 0:  7 14 15 -- 27 --
12:35:13.000          -8.0180
12:35:13.000 15 out  : 0:  7 14 -- -- 27 --
12:43:27.000          -1.6500
12:43:29.000          -1.5690
12:55:13.000 24 start: 0:  7 14 -- -- 24 27 --
12:58:09.000  9 start: 0:  7  9 14 -- -- 24 27 --
13:01:56.000 12 start: 0:  7  9 12 14 -- -- 24 27 --
13:15:13.000          -4.1200
13:15:13.000 27 out  : 0:  7  9 12 14 -- -- 24 -- --
```

Figure 18 – **jump** file

In the case of the jumps associated with satellites going out, corrections are applied by editing the **jump** file. A line is entered for each instance where a jump occurs that corresponds to a satellite going out. The entries are made in chronological order in the format DOY HH MM SS, with no colons. In the case of solutions that run over midnight, use caution to ensure that the proper day is entered. The **nav130** solution program will use these times and will make the appropriate corrections to the solution. In the example, there are four of these occurrences, at 11:01:20 , 12:21:07, 12:35:13 and 13:15:13. The entries appear as shown below, with a final 999 00 00 00 entry that is required to denote the end of the corrections.

267 11 01 20

267 12 21 07

267 12 35 13

267 13 15 13

999 00 00 00

All of the original lines in the **jump** file below the 999 00 00 entry should be deleted because it will be necessary to run **navjump** after each solution to determine if corrections have been applied correctly. Using the command:

% navjump >> jump

the results of subsequent solutions are added to the existing **jump** file. They can be examined in the same fashion and additional entries made at the top of the file as necessary. After one or two runs of **nav130** with a **jump** file, all of the jumps associated with satellites going out of the solution should be corrected.

The second primary source of jumps in the solution are cycle slips that have not been corrected by the initial editing. The **NAV22.EDT** file generated by **nav130** will usually contain the edits necessary to correct any remaining cycle slips, and it will frequently include edits that are not valid. An example NAV22.EDT file is shown in figure 19.

19	1	1	1	1	1720	999999	-5.0	10:31:39.00
15	1	1	1	1	1720	999999	-6.0	10:31:39.00
15	1	1	1	1	1721	999999	-6.0	10:31:40.00
15	1	1	1	1	1722	999999	-6.0	10:31:41.00
15	1	1	1	1	1723	999999	-5.0	10:31:42.00
15	1	1	1	1	1724	999999	-5.0	10:31:43.00
15	1	1	1	1	1725	999999	-5.0	10:31:44.00
13	1	1	1	1	2297	999999	47.0	10:41:16.00
14	1	1	1	1	3957	999999	8.0	11:08:56.00
14	1	1	1	1	3958	999999	8.0	11:08:57.00
14	1	1	1	1	3959	999999	8.0	11:08:58.00
14	1	1	1	1	3980	999999	5.0	11:09:19.00
14	1	1	1	1	3981	999999	5.0	11:09:20.00
14	1	1	1	1	3982	999999	5.0	11:09:21.00
14	1	1	1	1	3983	999999	5.0	11:09:22.00
14	1	1	1	1	3984	999999	5.0	11:09:23.00
14	1	1	1	1	3985	999999	5.0	11:09:24.00
14	1	1	1	1	3986	999999	5.0	11:09:25.00
14	1	1	1	1	3987	999999	5.0	11:09:26.00
19	1	1	1	1	8284	999999	-25.0	12:21:03.00

Figure 19 – NAV22.EDT File

The elements of the edit instruction are:

1. The SV PRN number
2. The beginning receiver
3. The ending receiver
4. The beginning frequency
5. The ending frequency
6. The beginning epoch
7. The ending epoch (999999 is used for end of file)
8. The size of the correction (or -9999.0 to delete)
9. The time of the correction (optional)

Each line in the edit file represents a single instruction which can be used by the program **cdata** to apply a correction to the database. The best way to determine which edit instructions are valid and should be applied to the data base is to compare the **NAV22.EDT** file with the **jump** file. In some cases, normally when there is a large cycle slip present, the **NAV22.EDT** file can contain thousands of invalid edits and it is not practical to do a comparison. In this case, the program **goodedits** will extract the edits that are likely to be valid from the **NAV22.EDT** file. By redirecting the output of **goodedits** using:

% **goodedits > good.edt**

the edits can then be more easily compared with the **jump** file. From the **NAV22.EDT** and **jump** files in the example , the only edit that corresponds to a jump occurs at 10:41:16. The other edits in the **NAV22.EDT** file are not valid.

Because it is prudent to save the original edit file in case it becomes necessary to restart the editing process, the **b1.edt** file should be copied to a new file before changes are made. By sequentially using a series of edit files, **b2.edt, b3.edt,** etc., it is easy to go back a step if a change has been made to the database that was incorrect and should be undone.

In the case of a single edit instruction, it is easiest to edit the **.edt** file and type the instruction in at the appropriate point. If there are numerous edits to be added, the command:

% **sort -n +5 good.edt b2.edt > b3.edt**

will create a new edit file with the instructions placed appropriately in the file.

The original **a267** database is always left protected and the working **c267** database is created by applying the current edit file to the **b267** database. Before applying the edits, the **HD.DAT, OR.DAT** and **AX.DAT** files are moved back to **b267** with the command:

% **mvv c b**

Then, **cdata** is used to apply the edits and create the updated **c267** database. After one or two iterations of running **nav130**, checking the **NAV22.EDT** file, adding any required edits to an updated **.edt** file and re-editing the database, all cycle slips that are detectable by **nav130** should be corrected.

The final test of the quality of the solution is examination of the solution position file, **PSOL267**. The PSOL plot file contains position information in XYZ and NEU (North East Up) formats. Running % **splot PSOL267** will display the plot menu, and selecting UP will display the plot shown in figure 20.

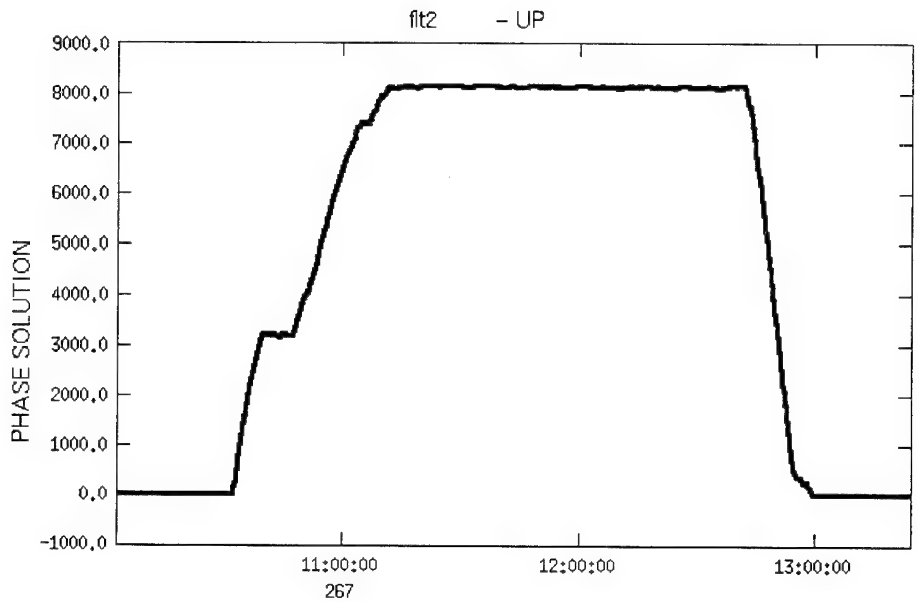

Figure 20 – PSOL UP Plot

If the aircraft returned to the same spot it started from, the pre- and post-flight UP values should agree closely . By using the mouse and expanding the PSOL plot during these periods, the UP values are compared.

Figures 21 and 22 show the UP data for the example and it can be seen that the post-flight UP position is within approximately 0.5 meter of the starting position (The starting value should always be 0.0). If the post-flight UP position is within a meter of the starting position, as is the case in the example, it is likely that all necessary corrections have been applied and the solution is of good quality.

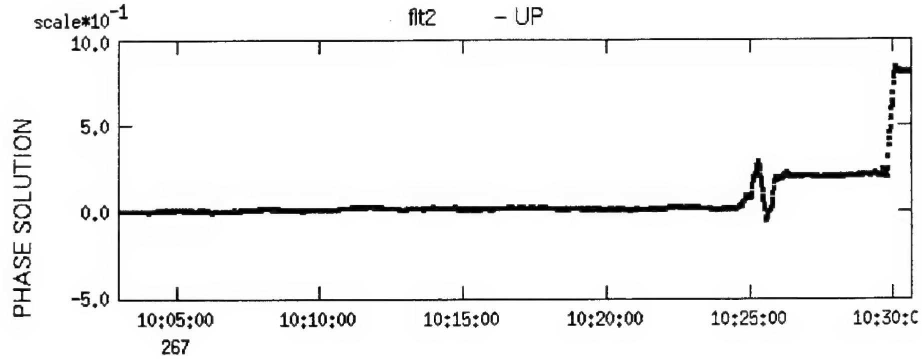

Figure 21 – Before Flight Aircraft Height

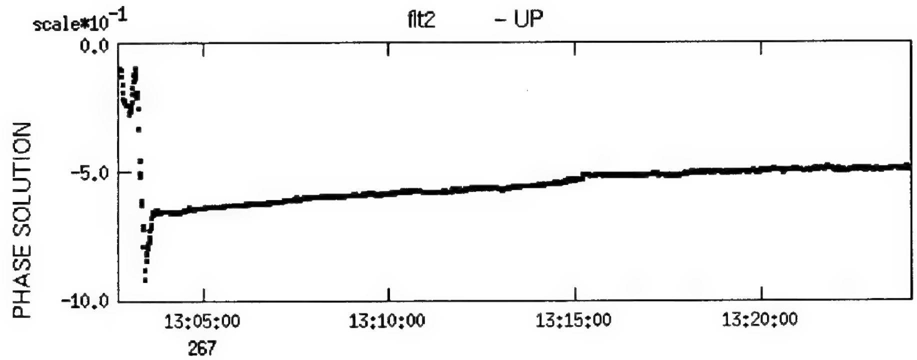

Figure 22 – Post Flight Aircraft Height

When a good solution has been obtained, the necessary files can be saved with the **svp** command. Because this was a forward L1 solution, % **svp f1** is used. This will move the **NAV22.OUT** solution file to **f1_psol.267**, and the **PSOL267** plot files to **f1_psol.LIM** and **f1_psol.PLT**. Also created are the files **f1_jump**, **f_hd.dat**, **f_gps22.inp** and **f_nav22.inp**.

Reverse Solution

The complete solution process consists of obtaining both the L1 and L3 (combined L1-L2) solutions in the forward and reverse direction. The forward and reverse solutions are then averaged to produce a final combined solution. It is a matter of personal preference whether to do the reverse L1(r1) or the forward L3(f3) solution after completing the forward L1 (f1), but for purposes of the example, the reverse L1 will be next.

The first step in setting up for the reverse solution is computing the post-flight static baseline. Even though the elevation of the aircraft will change very little, it is possible that the aircraft could be several meters away from the original starting position. Determine the time of the post-flight

static period and run **gpsset** to set the static solution parameters. In addition to entering the proper times in the **GPS22.INP** file, it may be necessary to select a different reference satellite. When the necessary entries have been made, exit **gpsset** and run **gps22e** to compute the baseline. The process of checking the baseline solution is the same as it was with the forward solution. Examine the **SAVIT** file and the DDR plot and make corrections as necessary. When a good solution has been obtained, save the aircraft receiver position data with the **savit** command.

The changes to the **NAV22.INP** file will normally be the same as with the **GPS22.INP** file. This will include entering the start and end times for the reverse solution and possibly a new reference satellite. The final step is removal of the forward **jump** file because the forward solution jump times are not valid in the reverse solution. When the setup is complete, run **nav130** to compute the reverse solution.

Making corrections to the reverse solution follows the same procedure as with the forward solution. First, run **navjump** to get the times for satellites that set during the solution and create the reverse **jump** file. The times will be listed in the same order that the solution is run - from later to earlier. The procedure for comparing the **NAV22.EDT** file with the **jump** file is also carried out the same way as with the forward solution. Because of the editing that was done during the forward solution, it is possible that no further editing will be required.

When the necessary corrections have been applied and examination of the PSOL plot indicates that a good solution has been obtained, save the solution using **svp r1**. This will create the files **r1_nav22.267**, **r1_psol.LIM** and **r1_psol.PLT,** and the corresponding **r_hd.dat**, **r1_jump**, **r1_gps22.inp** and **r1_nav22.inp** files.

Combining Solutions

Because any small, uncorrected cycle slips will gradually degrade the accuracy of a solution over time, the forward and reverse solutions are combined by averaging to minimize these errors. The process of combining solutions is interactive and the user must determine which portions of the solutions are to be averaged. Before the two solutions are combined, the reverse solution is transposed in time to coincide with the forward solution. The command

% mkb r1_nav22.267

creates the file **fr1_nav22.267,** which is the transposed reverse solution. Then, using the program **doavg**, the user interactively selects the region of the solution that is to be averaged. Run **doavg** with the forward solution and the transposed reverse solution using the command

% doavg f1_nav22.267 fr1_nav22.267

When **doavg** is run, it first decimates the solution files to a 30 second interval, and then cues the user to display the combined solution. By selecting **up**, a plot of the two solutions is displayed. Figure 23 shows the combined plot for the example. The first file entered for **doavg** will be plot **A** and the second file entered will be plot **B**. Normally, the forward solution is entered first and will be plot **A**.

Figures 24 and 25 are expanded views of figure 23 and depict the time periods when the aircraft is on the ground. As can be seen in figure 24, the forward solution height is steady and level at the beginning and the reverse solution is gradually decreasing towards it. The opposite situation can be seen in figure 25, where the reverse solution is the more steady.

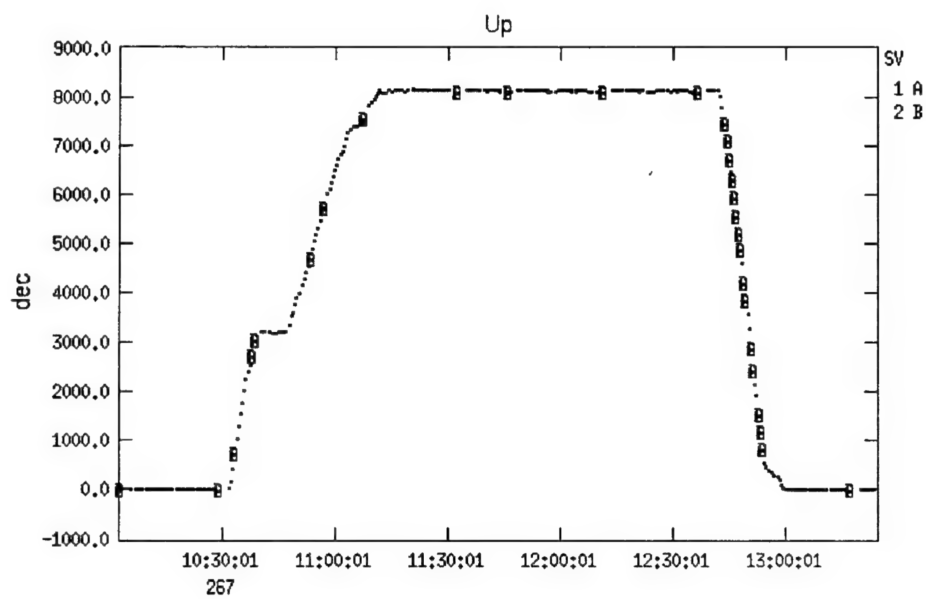

Figure 23 – Combined Solution Plot

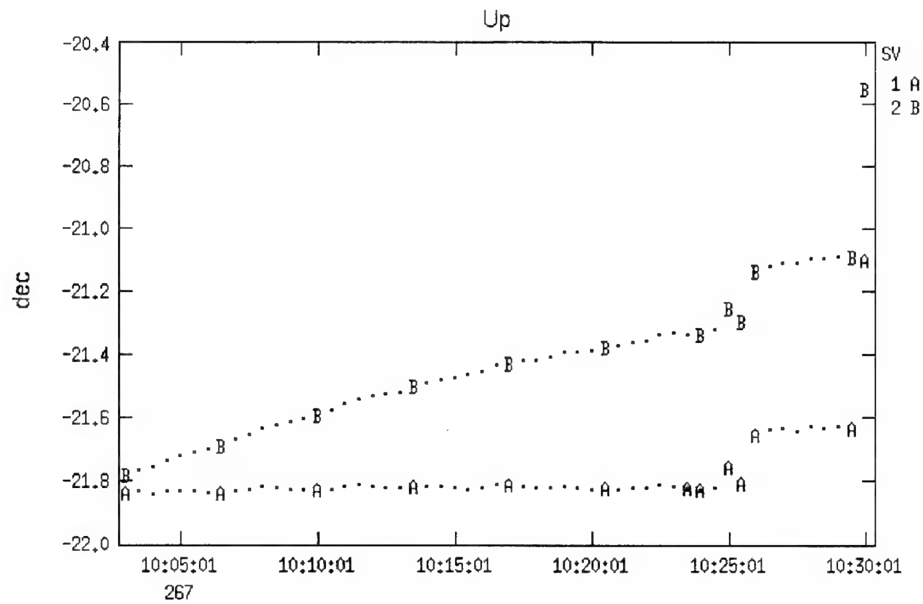

Figure 24 – Beginning of the Combined Solution

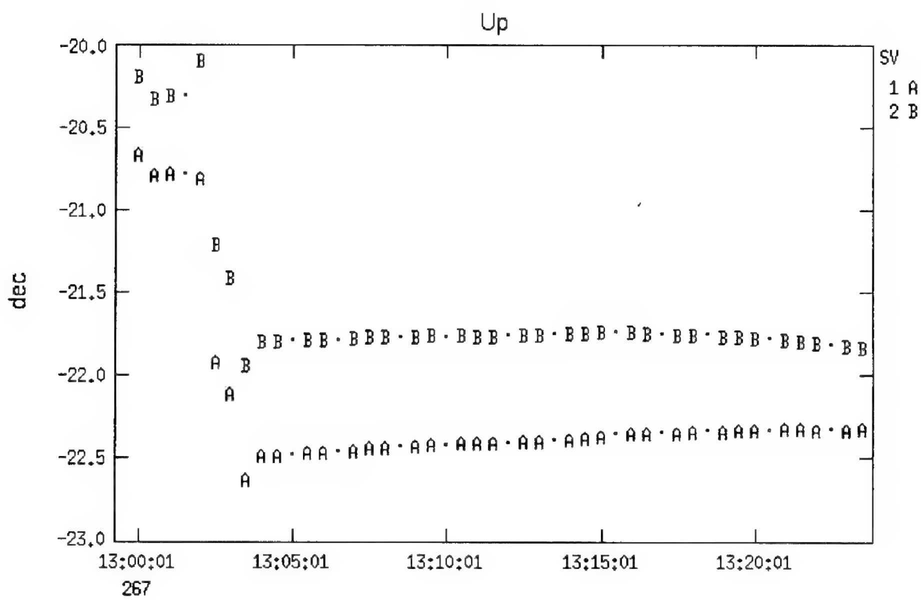

Figure 25 – End of the Combined Solution

Typically, there will be a least one point, and usually two, where the two plots cross over. By finding the times where the plots cross over and the vertical separation between them is minimum, two points can be selected for the start and end of the averaging. To avoid the possibility of a significant jump, the times should be a minimum of 30 minutes apart. When the times have been

selected, close out the combined plot and **doavg** will cue the user to enter them. They must be entered in the specific format **DOY HH:MM:SS.SS**.

After the times are entered, **doavg** will run and then cue the user to display a confirmation plot, which will show the original forward and reverse solutions plus the averaged data.

The confirmation plot can be examined to determine if a good averaging period was selected. If the result is satisfactory, close out the confirmation plot.

The combined solution file that is output by **doavg** will have the generic name **nav22.267**. Because this combined solution will be saved, it should be renamed to distinguish it from the other solutions already computed and others that will be computed later. The convention that is normally used is to use a letter **c** prefix to indicate the solution is combined, followed by a **1** or **3** to designate the frequency, and then a letter **b** or **p** if the broadcast or precise ephemeris was used for the solution. The complete name for this solution would be **c1p_nav22.267**.

The command

% **combnav c1p_nav22.267 | mknplt -n c1p_psol**

will create the plot files **c1p_psol.LIM** and **c1p_psol.PLT**.

Combined L3 Solution

The L3 solution combines L1 and L2 frequency carrier phase data. It will normally be the most accurate solution because the dual frequency data can be used to correct for certain atmospheric effects. Computing the L3 solution is similar to the process used to compute the L1.

The first step is to compute the edits for the L2 frequency and add them to those that have already been computed for the L1 solution. To do this, create a copy of the last edit file used. The naming convention is to give the new file the next higher number and to add a letter designator to clearly distinguish it from the L1 only edit files. If the last L1 edit file was **b3.edt,** use **b4l.edt** for the L3 edit file name. Next, edit the file **BDATA.INP** and change the third line to the new edit file name **b4l.edt**. Because the L2 edits are computed by **nav130** running in the forward direction, use **% mvp f1** to restore the forward solution **NAV22.INP** and database **HD.DAT** files. Then, run **% navset** to enter the NAV22 setup menu, set PROCESSING MODE to APRIORI and change FREQUENCY to L3AVE. As a final setup step, delete the current **jump** file. The L2 edits are computed by running the script **mkl2**, which sequentially runs **nav130**

to compute a solution and compute the L2 edits, adds the edits to the **b4l.edt** file, applies the new edits using **bdata** and then reruns **nav130**. The process ends when no cycle slips are detected and will usually take two or three iterations.

When the **mkl2** process is completed, run **% navset** to enter the NAV22 setup menu, change PROCESSING MODE to SOLUTION, and then run **% nav130.** When the forward L3 solution has run, enter % **navjump -3 > jump** to create the L3 jump file. As with the L1 solution, edit the **jump** file to enter the times when jumps occur that correspond to satellites that go out during the solution and rerun **nav130**. When all jumps are corrected and the final forward L3 solution is complete, run **% svp f3** to save the solution.

To do the reverse L3 solution, run **% mvp r1** to restore the reverse **NAV22.INP** and **HD.DAT** files, enter the NAV22 setup window with % **navset**, and change frequency to L3AVE. Remove the **jump** file from the forward solution then run **nav130**. Proceed as with the forward solution, using **navjump** to create the **jump** file, editing the **jump** file, and rerunning **nav130** as necessary until all jumps associated with satellites going out have been corrected.

To combine the forward and reverse L3 solutions, run the script % **docomb**, which will transpose the reverse solution, combine it with the forward solution, and then prompt the user to display the combined plot. As with the L1 solution, compare the two plots and determine the period for averaging. When the plot is closed enter the times chosen. When **docomb** has completed, **% svp r3** will move the **nav22.267** file to **c3p_nav22.267** and create the **c3p_psol** plot files.

Smoothed Pseudo-range Solution

When merge320 is run in the kinematic solution directory, it creates a smoothed pseudo-range solution file, **KINSLV.OUT**. This solution will normally be much noisier and significantly less accurate than the phase solutions, but it is created automatically and in some cases, it may not be possible to obtain phase solutions. The file **neu.raw** created by the command **kintoneu KINSLV.OUT > neu.raw** is an unfiltered north-east-up position file computed from p-code pseudo-range data. The command **newkin > neu.1** is then used to create a Doppler smoothed position file. Finally, using:

% **newkin -g 2.0 > neu.2**

will created a filtered version of **neu.1.** These three files plus one of the phase solution files are then combined into a single plot file with the command:

% **combnav -i 30 c3p_nav22.267 neu.raw neu.1 neu.2 | mknplt -n plot_name**

Run **xplot plot_name** to display the data files–they will be displayed in the plot with letters corresponding to the order they are entered in the **combnav** function.

DIFFERENTIAL GPS SOLUTIONS: BACKGROUND INFORMATION

Differential GPS

In the interferometric method, carrier-wave phase data from four or more satellites are simultaneously collected by two or more receivers. The unknown position of one receiver is determined with respect to that of the receiver with known location by solving a set of equations relating the receiver observables to the desired unknown quantities, Δx, Δy and Δz.

The measured phase observable in the GPS receiver is the difference between the phase of the receiver's local oscillator and the received phase of the L1 (1575.42 MHz, 19 cm wavelength) and/or L2 (1227.6 MHz, 24 cm wavelength) carrier-wave signals transmitted from a GPS satellite. The beat-frequency phase difference is accumulated from some initial point in time. This observable plus an unknown integer number cycles, N^k, may be thought of as a biased measurement of the distance from the receiver to satellite k. The range is biased by differences between the satellite and the receiver clocks and the signal propagation effects of the ionosphere and troposphere. A non-differential receiver position could be calculated if such measurements were made to four satellites with known positions and if each ambiguity, N^k ,could be determined. The fourth satellite observation is necessary to solve for the offset between the satellite and receiver clocks in addition to the three-dimensional position coordinates. Unfortunately, the abmiguities are not observable in this situation. Even if the ambiguities could be measured, the position would be in error by a few to tens of meters because of error in the satellite ephemerides and the unknown atmospheric effects.

The type of single receiver operation described above is used for positioning with the pseudo-range or code observable where the carrier wave transmissions are essentially tagged by pseudo-random binary codes modulated onto the carrier waves. Travel time measurements are made by means of the tags. The code transmitted at a known time from the satellite is correlated against a receiver generated copy of the same code. The time shift between the two codes is a measure of the range to the satellite plus atmospheric delay and receiver clock offset, hence the term pseudo-range. Since the estimated satellite ephemerides are also modulated onto the carrier signals, the geocentric position and clock offset of the receiver can be calculated from pseudo-ranges measured to four satellites at a single epoch.

The C/A, coarse acquisition, code is modulated onto the L1 carrier while both L1 and L2 are modulated by the P or precise code. The P-code chip rate is ten times greater than that of the C/A

code which allows finer time-resolution in the receiver correlation process. The different signal delays obtained on the L1 and L2 P-code pseudo-ranges can be used to correct the measurements for most of the ionosphere effects. Besides allowing the measurement of pseudo-ranges, the codes are also used in the phase tracking loops. A codeless method known as frequency-squaring is used to track the phase if the receiver is not capable of code correlation. However, this results in a phase-observable with half the wavelength of the actual carrier frequencies, a significant disadvantage for the processing of the phase data as will be seen below. The signal-to-noise ratio of codeless tracking is also much lower than code-aided tracking. This results in phase data with more noise and cycle-slips. A cycle-slip is the jump in measured phase that occurs when receiver phase-lock is lost for some period of time, effectively resetting the integer ambiguity. Cycle slips are caused by, e.g., an obstruction blocking the view of a satellite or a reduction in signal-to-noise below the tracking threshold. The types of GPS measurement observables available are then: L1 C/A-code pseudo-ranges, L1 P-code pseudo-ranges, L2 P-code pseudo-ranges and the phases of L1 and L2 tracked by coded and codeless methods. Relatively few models of GPS receivers are capable of measuring all of these observables. Receivers also vary in the number of satellites that can be tracked simultaneously, measurement rates and measurement noise and robustness.

Real-time, single receiver positions derived from dual-frequency P-code pseudo-ranges are normally accurate to 10-20 meters depending on satellite geometry. Typical single-frequency C/A-code position accuracy is 35-50 meters when not degraded by SA. Unfortunately for civilian users of GPS, the US Department of Defence has stated that the P-code will be encrypted for military use, and SA will reduce the accuracy of the C/A code positions to 100 meters once the system becomes fully operational. Several GPS manufacturers are working on methods of tracking the full wavelength L2 carrier-phase if the P-code is encrypted. The various methods result in some loss of signal-to-noise and the L2 pseudo-range will probably not be available. Even if the P-code were not to be encrypted, pseudo-range positions are not accurate enough for some positioning requirements. Instead, differential techniques between fixed and mobile receivers are used to reduce error sources such as satellite orbit and clock that are common to both receivers.

Differential pseudo-range positioning is significantly more accurate than single receiver operation and is relatively easy to accomplish. The pseudo-ranges of the mobile receiver are subtracted from those obtained from a fixed base-station receiver on a satellite by satellite basis. The only requirement is that the same satellites be observed at both locations and that the pseudo-ranges be time-tagged identically. The mobile position solution is then computed at each epoch relative to the base-station location from the delta-pseudo-ranges. This technique reduces not only the normal satellite clock and ephemeris errors, but also the increased clock and ephemeris errors

from SA. Ionosphere errors in single-frequency, C/A measurements are also reduced for base-lines of up to a few tens of km. in length. Differential pseudo-range positioning has proven so effective that several nations are establishing networks of base-stations around their coastlines. The base-station data will be broadcast for real-time use on ships and aircraft. For other applications it may be sufficient to record the data from the two receivers for post-processing. Improvement in accuracy from about 100 meters to 5 meters has been reported in shipboard tests for base-lines of up to 200 km. The mobile antenna installation is a critical issue in this type of positioning as pseudo-range measurements are subject to errors caused by multi-path interference. Multi-path errors are not related between the fixed and mobile receiver and are not removed by differencing. Unfortunately, it has proven difficult to eliminate the effects of multi-path on aircraft due to the large number of convex reflecting metal surfaces in the area of the antenna.

Static Interferometric GPS

Differential phase, also known as interferometric-mode GPS, is much less subject to multi-path interference. The effects are reduced from meters to centimeters. Over baselines of 10-70 km accuracies of about 1 ppm in the horizontal and 2-3 ppm in the vertical are commonly achieved in static interferometric positioning (Schwarz et al., 1987). Neglecting atmospheric effects, the carrier-phase observable as a function of receiver time is given as (Leick, 1990):

$$\begin{aligned}\Phi_k^p(t_r) &= a^p(t-t_0)+\frac{1}{2}b^p(t-t_0)^2+\Phi_T^p(t_0)-\frac{f}{c}\rho_k^p(t) \\ &\quad -[a^p+b^p(t-t_0)]\frac{\rho_k^p(t)}{c}-\frac{f}{c}\dot{\rho}_k^p(t)dt_k-\Phi_k(t_0)-fdt_k+N_k^p\end{aligned} \tag{1}$$

where

t_r, dt_k, t, t_0 = receiver time, clock error of receiver k, true time and reference epoch and $t_r = t + dt_k$,

f, a^p, b^p = the nominal value, offset and drift rate of the frequency from satellite p,

$\Phi_T^p(t_0)$ = the phase transmitted from satellite p at the reference epoch,

$\Phi_k(t_0)$ = the oscillator phase of receiver k at the reference epoch,

$\rho_k^p(t), \dot{\rho}_k^p(t)$ = the topocentric range and range-rate between receiver k and satellite p and

N_k^p = the integer ambiguity between receiver k and satellite p.

A heirarchy of so-called single-, double- and triple-differences provides various advantages in observation and calculation. As is the case with differential pseudo-ranges, all of the differential phase combinations greatly reduce the effect of ephemeris errors on the position solution. The reduction is a function of the baseline length between receivers, scaling roughly as the ratio of the baseline length to the topocentric satellite distance multiplied by the satellite orbit error. The component of position error from typical precise-ephemeris (the post-processed orbits released a few weeks after real-time) errors of five meters and baseline lengths of 20 km. is on the order of .5 cm. This increases to 25 cm. at station separations of 1000 km. These position errors can be reduced by determining an improved satellite ephemeris that incorporates data from several stationary receivers in the survey region in addition to the main tracking sites.

A single-difference is calculated as the difference in phase observed at the reference and unknown receivers from a given satellite. This cancels the unknown initial transmitted phase term in equation 1 and eliminates first-order effects of the satellite clock error since both parameters are common to each receiver. A smaller clock error term that scales with the baseline length remains. The subtraction (single-difference) of the two individual receiver-satellite ambiguities produces a new combined integer ambiguity parameter and lumps the two receiver clock errors together as the difference of the two errors.

The next level is refered to as a double-difference, where two pairs of single-differences between two receivers and two satellites are subtracted to remove the first-order receiver clock error in addition to the error reductions obtained by the single-differences. The double-difference integer ambiguity parameter is formed by the analogous subtraction of the single-dfference ambiguities.

A triple-difference is the difference over time of a pair of double-differences from different epochs. Assuming no cycle slips occur, the integer ambiguity terms cancel because the ambiguities are constant in time. Triple-differences also have the previous advantages gained by the other combinations.

Obviously, no new information has been gained in computing double-differences from the previously formed single-differences at a given epoch. All of the information necessary to determine the receiver clock errors that cancel in the double-difference combination is available from the single-differences. Only three independent double-differences can be calculated at each epoch from the four independent single-differences, given observations of four satellites from two receivers. In exchange, one less parameter per epoch, the receiver clock error, must be estimated from the double- vs. single-differences. The integer phase ambiguities are global for a sequence of

epochs with no cycle slips and must be determined for either of these combinations to calculate the unknown station position. The position parameters, Δx, Δy and Δz between the reference and unknown station are also global over a sequence of observations if the two receivers are stationary. A set of underdetermined equations in the observations, the position deltas, ambiguity and station clock (for single-difference observations) parameter unknowns can be formed for each observation epoch. Although observations spaced closely in time are highly correlated, the system becomes overdetermined after some period of time due to the changing geometric constraints of the satellites' orbital motion and can be solved for the unknown global parameters. The length of time required depends on the number (minimum four) and geometry of satellites observed, station separation and the amount of a priori or additional information available to constrain the system. For example, the receiver clock corrections can be determined from a four satellite pseudo-range solution and used to reduce the order of the system. More numerous satellites, P-code pseudo-ranges and dual-frequency phase observations can all provide more stringent constraints and hence shorter observation periods. The minimum time for ambiguity resolution under these circumstances seems to be on the order of five to ten minutes. The so-called "rapid-static" observation method depends on the use of such redundant data to minimize the observation period required.

The triple-difference automatically incorporates new data over time. This is what allows the cancellation of the ambiguities. Over short periods the triple-difference position solution is weak, providing poor position accuracy. A single- or double-difference position solution over the same period would not resolve the ambiguities correctly, also providing a poor position. All of the procedures are equivalent when correlations between observations are accounted for (Leick, 1990). However, the methods provide varying levels of computational convenience under different circumstances.

As mentioned above, the ionosphere affects the propagation of the GPS signal. The carrier-phase is advanced and the modulated codes are delayed as compared to the same signal in a vacuum. The effect is a function of the signal frequency and the total (free) electron content (TEC) along the propagation path. A model ionosphere is used for corrections if only C/A-code pseduo-ranges are available. However, about 3-5 meters of unmodeled range error to the satellite remain (Klobuchar, 1986). The primary purpose for transmitting P-code modulations on two frequncies is so that military GPS receivers can easily determine a correction for ionospheric delay in the pseudo-range position solution. A simple first-order correction for the delay, accurate to about 5 cm., is used in the following expression (Leick,1990) for the topocentric range from satellite to receiver:

$$\rho_k^p = \frac{f_{L1}^2}{f_{L1}^2 - f_{L2}^2} P_{k,L1}^p - \frac{f_{L2}^2}{f_{L1}^2 - f_{L2}^2} P_{k,L2}^p \tag{2}$$

where
ρ_k^p = topocentric distance from receiver k to satellite p, uncorrected for receiver clock offset,
f_{L1}, f_{L2} = the L1 and L2 carrier-frequencies and
$P_{k,L1}^p, P_{k,L2}^p$ = the pseudo-ranges measured on L1 and L2.

The analogous expression for range in terms of the dual-frequency phase is:

$$\rho_k^p = \frac{c}{f_{L1}}[\Phi_{L1}^p - \Phi_{k,L1} - a\Phi_{k,L1}^p - b\Phi_{k,L2}^p + aN_{k,L1}^p + bN_{k,L2}^p] \tag{3}$$

where

$$a = \frac{f_{L1}^2}{f_{L1}^2 - f_{L2}^2} \text{ and}$$

$$b = \frac{f_{L1} f_{L2}}{f_{L1}^2 - f_{L2}^2}.$$

While ionosphere delay for a satellite may be computed directly from dual-frequency pseudo-ranges, the phase advancement is only available after a solution resolving the integer ambiguities has been calculated. Unfortunately, the relatively short horizontal correlation length of TEC profiles through the atmosphere greatly impacts the ability to resolve the integer ambiguities. If two receivers are only a few kilometers apart the receiver-receiver differencing of the phase data largely eliminates the ionosphere error, because the signal path from the satellites to each of the receivers is almost the same. Phase-bias resolution in a two receiver network is difficult or impossible beyond baselines of a few tens of kilometers because the signals traveling between the satellites and the two receivers experience different TEC along the paths and, hence, do not cancel. Also, the delays determined from dual-frequency pseudo-ranges contain too much receiver and multi-path noise to allow the fixing of integers in the current generation of receivers and antennas. In general, the ambiguities in long baseline interferometric GPS must be determined in the solution as real numbers ("floating biases") over the solution period. As mentioned above, long baselines also increase the component of ephemeris error in the solution, adding to the difficulty of resolving the ambiguities. Good positions can be obtained, even with floating biases,

but require long observation periods to average out the effects of the ionosphere and ephemeris error.

The above discussion assumed no cycle-slips in either receiver, a highly unrealistic assumption. Even small cycle slips must be fixed to achieve an accurate position solution. Repair of cycle-slips generally involves the examination of a parameter that will remain fairly constant over time except at slips including: triple- and double-differences and their associated solution phase-residuals; range-residuals; and the ionosphere-residual parameter.

A triple-difference time-series that contains a cycle-slip shows a spike at the time of the slip, while the double-differences have a step change at the slip. A position solution calculated from the differences will have residuals at each epoch between the solution and the differences. The character of the double- and triple-difference residuals is similar to that of the differences. The size of the spikes and steps may be used to estimate the size and then to correct the slips to some degree of accuracy depending on the other parameters of the solution. However, with two receivers observing, the slip could be in either of the receivers.

A range-residual is the difference in satellite-receiver range between the range computed by pseudo-range and phase methods. Since pseudo-ranges are immune to cycle-slips, any such slips show up as jumps in this difference. The single-frequency range-residual difference over time is given by:

$$\Delta\rho_k^p(t) = \lambda[\Phi_k^p(t) - \Phi_k^p(t_0)] - [P_k^p(t) - P_k^p(t_0)] \qquad (4)$$

The range-residual is independent of clock and troposphere errors. However, there is a considerable amount of noise in this parameter because of multi-path and receiver measurement noise on the pseudo-ranges. The receiver noise dominates since multi-path tends to be well correlated over short time periods, . Early P-code receivers had pseudo-range measurement noise of several meters. However, some newer receivers are claiming measurement noise of a fraction of a meter. In any case, filtering of the range residuals can help in the discrimination of cycle slips. High sampling rates also improve the chances of correcting cycle-slips by this method.

The ionosphere-residual parameter provides an excellent method for correcting slips in dual-frequency data, assuming that the L1 and L2 phase tracking-loops are independent, i.e. that both frequencies do not slip the same number of cycles at the same epoch. Goad (1985) gives the ionosphere-residual as:

$$\Delta_{ion}(t) = \Phi^{p}_{k,L1}(t) - (\frac{f_1}{f_2})\Phi^{p}_{k,L2}(t) \tag{5}$$

The ionosphere-residual measures receiver phase-measurement noise plus higher-order terms in the ionosphere propagation that are well correlated over distance. Its value tends to be a small fraction of a cycle over short time periods unless a cycle-slip occurs. A slip shows up as as a step in the series with the step size being a linear combination of the change in the integer ambiguity on the L1 and L2 phases. The size of the slips must be known to within a few cycles before attempting to correct simultaneous slips on both frequencies by this method as the linear combinations become ambiguous outside this range. That is, nearly the same size jump can be produced by different combinations of L1 and L2 slips above a few cycles. However, the method can find jumps of any size in one frequency, if it is known that there are no cycle slips on the other frequency. This is useful to find slips in the L2 data once the L1 phases have been edited by other means.

Kinematic Interferometric GPS

In the kinematic version of the static interferometric technique, one of the receivers is carried aboard a moving vehicle. The goal is to calculate a trajectory that is accurate to a fraction of a GPS carrier wavelength at rates of 1 to several Hz. However, the difficulties of the static method are compounded by additional difficulties in ambiguity resolution, more numerous cycle slips that are much harder to repair, and very large data sets. Even under perfect conditions kinematic GPS positioning is somewhat less accurate than static GPS because a new position must be calculated for each epoch, rather than averaging data over some tens of minutes for a single position. That is, the position deltas are no longer global to the set of observations, adding three more unknowns to the system for each epoch. With sufficient redundant data over time it is possible to resolve the integer ambiguities, even while in motion, by means of an ambiguity function technique (Mader, 1990) or Kalman filtering (Landau, 1988). However, the unknown ionospheric delays have so far limited such kinematic bias resolution to baselines of a few tens of kilometers. Therefore, the biases must, in general, be determined during stationary data collection periods near the base-station and carried into the kinematic period. Since the kinematic solution can be worked both forward and backward in time it is advantageous to collect stationary data both before and after the kinematic period. The biases determined during pre- and post-survey static periods should be identical if the same satellite constellation is visible throughout the entire period and no cycle slips occur on either receiver. Unfortunately, the requirement to observe the same satellites before, during and after the kinematic period would limit the duration to no more than two or three

hours. Surveys can be extended through two satellite constellations, perhaps five to six hours, by fixing the biases for the first constellation before the flight and propagating the position solution forward in time until the first set of satellites sets and the second set rises. The biases for the second set are determined after landing and the solution worked backward in time to meet the forward solution in the middle of the flight.The two solutions connect at the accuracy limit of kinematic measurements, assuming again no cycle slips in either receiver.

It may, on occassion, be neccessay because of receiver failure or extended kinematic periods during a collection to use some satellites that rise after the start of the kinematic period and set before the end. These satellites require kinematic phase-bias initialization. The method of initialization used in XOMNI is to select the integer bias for the new satellite that best fits the new data to the current position at that epoch. This is done by minimizing the phase-residuals of all double-difference combinations involving the new satellite to the a priori position estimate calculated without the new satellite. Such kinematic initializations are inevitably in error by one or more cycles due to: the unknown ionosphere delay along the path to the new satellite; the increasing effect of orbital error with baseline length; and the change in solution volume related to the changed geometric error with the new satellite. Research on multi-epoch bias estimation over long base-lines is currently in progress to try to improve on the single epoch method described here (Columbo,1992; Columbo and Peters, 1992).

Any incorrectly estimated phase-biases produce a position error that tends to grow with time. This reduces the accuracy of any subsequent kinematic phase-bias estimates, causing additional growth in position error over time. The average position error is reduced by calculating the solution from both ends of a survey towards the middle, but can exceed 10-20 meters near the middle of a 10 hour survey. The error is relatively smooth in character and of long wavelength and so can be ignored, or compensated for with some applications. The cycle-slip situation is also worse for the kinematic case than for static positioning. Receiver dynamics require greater bandwidth in the phase-tracking loops which reduces the signal-to-noise ratio. This causes more cycle slips and greater phase noise than the static case. The extended observation periods required for long-range surveying also necessitate the collection of some data from low elevation satellites, again increasing the number of cycle-slips and the amount of phase noise. In practice, the goal must be to minimize the number of cycle slips and their effect by careful operational methods and then to correct during the post-processing of the data whenever possible the cycle-slips that do occur. Methods for minimizing cycles-slips and increasing the chances of repair include: use of P-code receivers known to have robust phase- tracking characteristics; scheduling flights during periods of good satellite geometry with more than four satellites at elevations above 20 degrees;

collecting data from more than one GPS receiver on the vehicle and at the base station; proper selection of antenna and attention to mounting position to minimize multipath interference and wing and empennage obstruction during aircraft maneuvers; and minimizing bank and climb angles during aircraft operation.

Several techniques may be used to estimate corrections for most relatively short period cycle-slips which unavoidably occur in kinematic data sets. These include: delta-range trend comparisons of P-code pseudo-ranges to phase derived ranges (Mader, 1986); integer ambiguity search to minimize the change in the dual-frequency ionospheric residual parameter across a slip (Goad, 1985); P-code wide-lane/narrow-lane search (Blewitt, 1990; Landau, 1988); and short period navigation from raw inertial data (Eissfeller and Spietz, 1989; Hein et al., 1989, Brozena et al., 1989). These techniques are capable of spanning gaps of a few to a few tens of seconds in the phase record. Their applications variously require P-code pseudo-range, dual-frequency phase or inertial data. Since the phase solution can be worked forward or backward in time, a single large data gap may also be spanned by forward computing from the onset of movement to the gap and backward computing from the end of the kinematic period to the gap.

Multiple GPS Receiver Methods

The use of multiple receivers and antennas increases the probability of maintaining phase lock on at least 4 satellites even though the data may be acquired by different receivers. In such a case, the phase data must be be geometrically moved to a common location through the use of attitude data and the fixed relationship of antenna locations. Furthermore, if two or more receivers were operated from a single antenna, those observations would be stationary with respect to each other regardless of vehicle motion and could be examined using the double-difference techniques commonly used in static GPS work (Mader et al., 1991).

Carrier-phase cycle-slips are readily detectable in short base-line static GPS data through examination of the double-difference phase residuals. Given two GPS receivers observing the same two satellites at a given epoch, a simplified double-difference phase-residual (see Apendix 1 for the full expression) can be defined (Mader, 1986) as

$$\Delta\Phi_t = (\Phi_k^p - \Phi_{k_0}^p) - (\Phi_k^{p_0} - \Phi_{k_0}^{p_0}) \tag{6}$$

where Φ_k^p denotes the carrier phase observable for SV p from receiver k, and the zero subscripts indicate the reference receiver and satellite. Given good positions for the static sets,

accurate satellite ephemerides and cycle slip-free observations, a plot of the difference versus time should be flat and continuous. Also, GPS receiver manufacturers routinely test their equipment by attaching two or more receivers to a single antenna and then comparing the differences between sets. This technique can be extended to the problem of detecting cycle slips in kinematic receivers by simulating the static situation and using multiple kinematic receivers in a fixed configuration on the observing vehicle.

GPS Double-Difference Phase Equations

Mader (1986) derive the following equations used in the OMNI and XOMNI kinematic GPS reduction software. The notation has been changed to that of Leick (1990) to be consistent with the equations in the text. The phase observable at receiver k from a satellite p is:

$$\Phi_k^p(t_R) = \Phi^p(t_T) - \Phi_k(t_R) \tag{7}$$

where
t_T = transmission time,
t_R = reception time,
Φ^p = phase transmitted from satellite p and
Φ_k = phase of the local oscillator of receiver k.

Expressing the transmission time in terms of the reception time,

$$t_T = t_R - \rho_k^p(t_T, t_R) / c = t_R - \Delta t \tag{8}$$

where
$\rho_k^p(t_T, t_R)$ = the distance travelled by the signal from satellite p at time t_T, to receiver k at t_R,
c = the speed of light and
Δt = the light travel time.

Substituting equation (2) into equation (1):

$$\Phi_k^p(t_R) = \Phi^p(t_R - \Delta t) - \Phi_k(t_R) \tag{9}$$

Since the light travel time is small, the transmitted phase may be expanded around the received time:

$$\Phi^p(t_R - \Delta t) = \Phi^p(t_R) - \Delta t d\Phi^p / dt \tag{10}$$

where
$d\Phi^p / dt = f^p$, the transmission frequency of satellite p. Substituting in equation 9,

$$\Phi_k^p(t_R) = \Phi^p(t_R) - (f^p / c)\rho_k^p(t_T, t_R) - \Phi_k(t_R) + N_k^p \tag{11}$$

where N_k^p =an integer phase bias introduced since the absolute phase difference is not observed.

The next step is to expand the reception time in terms of GPS time, t_G, which is offset by a small amount from the receiver oscillator time, $t_R = t_G + \tau_k$:

$$\Phi_k^p(t_R) = \Phi^p(t_G) + f^p\tau_k - (f^p / c)\rho_k^p(t_T, t_G) - (f^p / c)\dot{\rho}_k^p(t_T, t_G)\tau_k - \Phi_k(t_G) - f_k\tau_k + N_k^p \tag{12}$$

where $\dot{\rho}_k^p(t_T, t_G)$ = the range rate and,
$f_k = \dot{\Phi}_k(t_R)$, the receiver oscillator phase rate of change.

A calculated or model phase observable is introduced in order to later obtain a linearized equation for the unknown station coordinates. The model phase is given by

$$\Phi_k^p(t_G)_{\text{model}} = (f / c)\rho_k^p(t_T, t_G)_{\text{model}} + (f / c)\rho_{\text{TROP}} \tag{13}$$

where
ρ_{TROP} = tropospheric delay correction.

The range is obtained by iterating the following expression. The model range for equation 13 results from the use of a priori estimates of the unknown station coordinates and velocities.

$$\begin{aligned}\rho_k^p(t_T, t_G) = [&(x^p(t_G) - \dot{x}^p(t_G)\rho_k^p(t_T, t_G) / c - x_k)^2 + \\ &(y^p(t_G) - \dot{y}^p(t_G)\rho_k^p(t_T, t_G) / c - y_k)^2 + \\ &(z^p(t_G) - \dot{z}^p(t_G)\rho_k^p(t_T, t_G) / c - z_k)^2]^{1/2}\end{aligned} \tag{14}$$

where
$x^p, y^p, z^p, \dot{x}^p, \dot{y}^p, \dot{z}^p$ = the earth-centered,earth-fixed coordinates and velocity components of

satellite p and
x_k, y_k, z_k = the coordinates of receiver k.

Subtracting equation 13 from equation 12 yields the residual phase,

$$\begin{aligned}\Delta\Phi_k^p &= \Phi^p(t_G) + f^p\tau_k - (f^p/c)\Delta\rho_k^p(t_T,t_G) - (f^p/c)\dot{\rho}_k^p(t_T,t_G)\tau_k - \Phi_k(t_G) - f_k\tau_k + N_k^p \\ &\cong \Phi^p(t_G) - (f^p/c)\Delta\rho_k^p(t_T,t_G) - (f^p/c)\dot{\rho}_k^p(t_T,t_G)\tau_k - \Phi_k(t_G) + N_k^p\end{aligned} \tag{15}$$

because the satellite and receiver phase offset terms are nearly equal.A reference receiver, k_0, and a reference satellite, p_0, are introduced in order to form the double-difference phase-residual combination,

$$\begin{aligned}(\Delta\Phi_k^p - \Delta\Phi_k^{p_0}) - (\Delta\Phi_{k_0}^p - \Delta\Phi_{k_0}^{p_0}) = &-(f/c)(\Delta\rho_k^p - \Delta\rho_k^{p_0} - \Delta\rho_{k_0}^p + \Delta\rho_{k_0}^{p_0}) \\ &-(f/c)(\tau_k(\dot{\rho}_k^p - \dot{\rho}_k^{p_0}) + \tau_{k_0}(\dot{\rho}_{k_0}^p - \dot{\rho}_{k_0}^{p_0})) \\ &+N_k^p - N_k^{p_0} - N_{k_0}^p + N_{k_0}^{p_0}\end{aligned} \tag{16}$$

where

$$\Delta\rho_k^p = (1/\rho^p_{k_{model}})[(x^p - x_k)\Delta x_k + (y^p - y_k)\Delta y_k + (z^p - z_k)\Delta z_k] \tag{17}$$

is the range-residual or difference between the true range and model range obtained from the a priori estimate of the receiver location. The time index has been dropped for brevity. Since the reference receiver is stationary and the location is known, the range residuals involving receiver k_0 may be assumed to equal zero. The Doppler terms of Equation 15 may be calculated from the range-rate obtained using estimated receiver positions. Also, assuming no cycle slips, the initial integer phase biases can be eliminated from equation 16 by subtracting the phase-residual double-difference at the first epoch of measurement.

Then

$$\begin{aligned}&[(\Delta\Phi_k^p - \Delta\Phi_k^{p_0} + (f/c)\tau_k(\dot{\rho}_k^p - \dot{\rho}_k^{p_0})) \\ &-(\Delta\Phi_{k_0}^p - \Delta\Phi_{k_0}^{p_0} + (f/c)\tau_{k_0}(\dot{\rho}_{k_0}^p - \dot{\rho}_{k_0}^{p_0}))]_t \\ &-[(\Delta\Phi_k^p - \Delta\Phi_k^{p_0} + (f/c)\tau_k(\dot{\rho}_k^p - \dot{\rho}_k^{p_0})) \\ &-(\Delta\Phi_{k_0}^p - \Delta\Phi_{k_0}^{p_0} + (f/c)\tau_{k_0}(\dot{\rho}_{k_0}^p - \dot{\rho}_{k_0}^{p_0}))]_{t_1} = -(f/c)(\Delta\rho_k^p - \Delta\rho_k^{p_0})_t + (f/c)(\Delta\rho_k^p + \Delta\rho_k^{p_0})_{t_1}\end{aligned} \tag{18}$$

If the initial position of the roving receiver is known, the initial range residuals can be set to zero. Writing the left side of Equation 11 as Φ,

$$
\begin{aligned}
(c/f)\Phi = & -((x^p - x_k)/\rho_k^p - (x^{p_0} - x_k)/\rho_k^{p_0})\Delta x_k \\
& -((y^p - y_k)/\rho_k^p - (y^{p_0} - y_k)/\rho_k^{p_0})\Delta y_k \\
& -((z^p - z_k)/\rho_k^p - (z^{p_0} - z_k)/\rho_k^{p_0})\Delta z_k
\end{aligned} \tag{19}
$$

This is the single-frequency phase equation for one double-difference used in the OMNI and XOMNI software. Three such double-difference equations from four satellites observed by both receivers are are required to solve for the delta x,y and z, which are the corrections to the assumed position of the unknown station. Additional satellites provide redundancy and allow a least-squares solution for the position deltas.

TECHNICAL DOCUMENTATION

The following text is a procedure level description of the operations carried out by the various XOMNI software modules. Most of the modules read a setup file to determine various parameters. These files are described in the appendixes. Each parameter has been assigned a name which can be used to refer between the module description and the setup file description.

MERGE

Main program

Figure 1 shows the overall execution flow for MERGE.

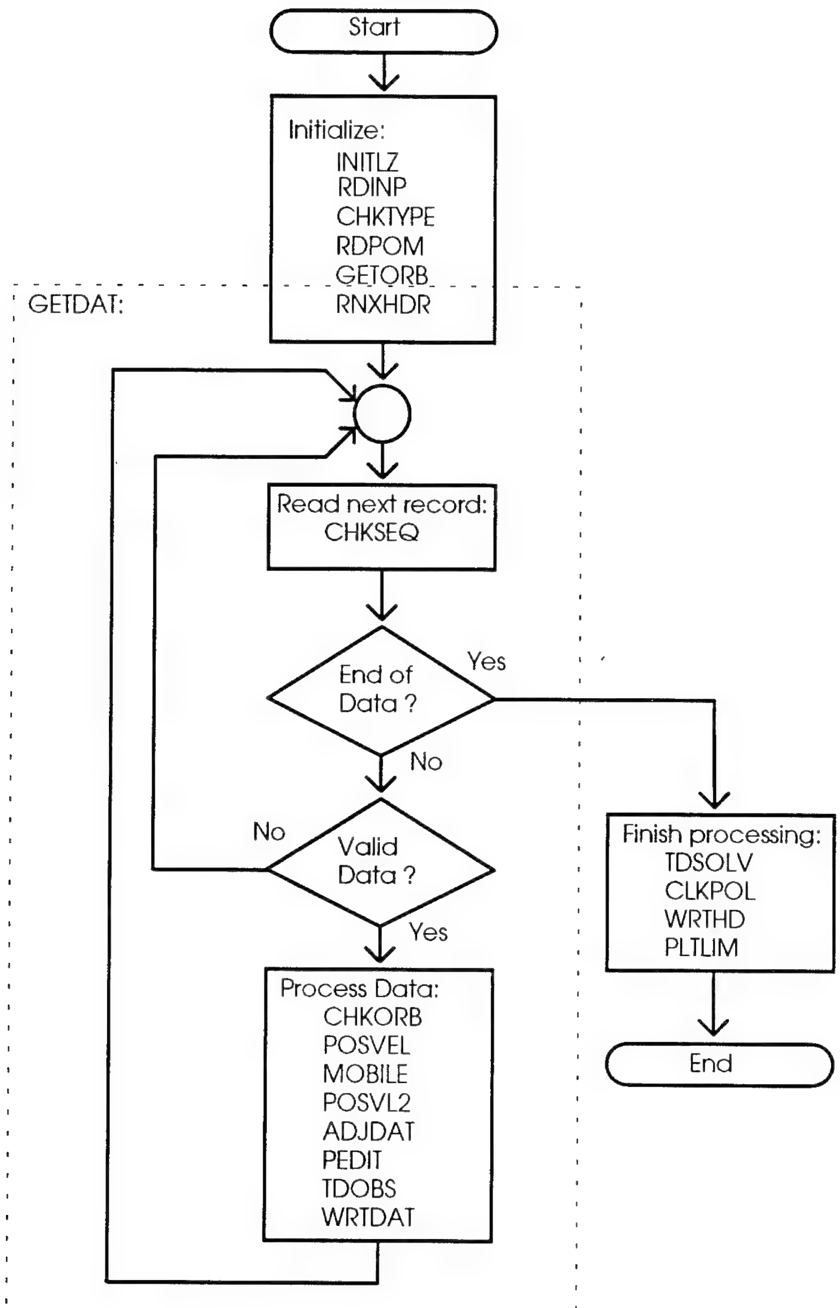

Figure 1 – MERGE Flowchart

The MERGE procedure simply calls several other procedures to acomplish the MERGE processing (note that the main loop occurs within GETDAT). These procedures and their functions are:

INITLZ is called to initialize all variables

RDINP is called to read the merge setup file

CHKTYPE is called to open check the type of the ***.DAT** files

RDPOM is called to read the ***.POM** files

GETORB is called to read the broadcast orbit file

GETDAT is called to read and process data epoch-by-epoch

TDSOLV is called to compute triple difference solutions

CLKPOL is called to solve for the clock polynomials using the pseudo-ranges

WRTHD is called to Create the database header file (**<db>HD.DAT**)

PLTLIM is called to write the plot-limit files (***<doy>.LIM**)

ADJDAT: Adjust phase and range to master clock time

Solve for Station Clock Polynomials

For each valid station SIMEQ is called to solve for the station clock polynomial coefficients using the normal matrix loaded by PEDIT. The parameters passed to SIMEQ must be set depending on how many observations are available. This determines the order of the polynomial.

Adjust the Phase and Range Data.

The adjustment is simply the computed time derivative of the value multiplied by the time offset. For RINEX files the time derivative is taken directly from the input file. For NGS files it is calculated from the radial velocity between the satellite and the station computed in RESID and the clock polynomial computed above. The time offset is the difference between the data time tag and the master clock corrected by the receiver clock error computed from the pseudo-range residual.

Recompute all Phase Residuals

RESID is called to recompute the phase residuals

ARCFIT: Compute polynomials for precise orbit

Read precise orbit file and write to scratch file

This is only done the first time ARCFIT is called. The input precise orbit file is an ASCII file for portability. The data is read in and written to a scratch binary file for faster access. Only the records required for the current processing run are saved. If the required data is not in the file a message is displayed and the program is terminated.

Read scratch file and form partials and normal matrix

The precise orbit data, which consist of a position and velocity recorded every 15 minutes, is read from the scratch file. 15 of these records are read at a time and fit to a 12 degree polynomial. The fit is performed by calling SIMEQ.

Compute Residuals of Fit

The residuals are computed as the difference between the input positions and velocities and the polynomial values.

Record ARCFIT.OUT

The original precise orbit data, the polynomial coefficients, and the residuals are output to the ARCFIT.OUT file.

BCEPH: Compute Position and Velocity from Broadcast Ephemeris

This procedure is called once for each satellite to compute its position and optionally its velocity. This calculation is straightforward and well documented in the literature (DMA, 1987; Van Dierendonk et al., 1980).

BCORB: Select Broadcast Ephemeris Elements Nearest Current Time

Search for nearest times

The broadcast ephemeris data for satellite passed in is checked to find the record before and after the current time.

Load elements and call BCEPH

The selected broadcast ephemeris elements are read from the array and stored for use by BCEPH. BCEPH is then called to compute the satellite position for the current time base on these parameters. The next set of parameters are then loaded and BCEPH is called again.

Calculate weighted average of positions and velocities

The weighted average is calculated based on the times of the ephemeris elements vs. the current time.

BCREAD: Read .ORB file and Store Broadcast Ephemeris Elements

Read header and identify file type

The input broadcast ephemeris file can be in either NGS (Chin, 1989) or RINEX (Gurtner and Mader, 1990) format. The first line of the file is read to determine the version.

Loop over file to read all records

A record header is read from the file. The record data is then read using a format appropriate for the file version. The time tag of the record is checked to make sure it is within the time range of the processing run. The satellite clock data of the record closest to the middle of the processing times are stored for use in modeling the satellite clock. The valid records are stored for use by BCORB. The next record is read and the process repeats.

CHKORB: Check if New Orbit Polynomials are Needed

Convert time

The master clock time, which is stored as fractional days of the year, is converted to various other time values for use in other procedures. These are: integer day of year, integer day of GPS

week, hours, minutes and seconds of day, day of month, month of year, and seconds of GPS week. The epoch count in also incremented.

Check Orbit time

If precise orbit data is begin used the current time is checked against the last time the current precise orbit polynomials are valid. If it is passed this time then ARCFIT is called to generate new orbit polynomials and the last valid time is reset.

CHKSEQ: Check if new station data is required

For each station CHKSEQ checks the time tag from the last record read from the input data file against the master clock time. If the data file is behind, records are read until it catches up. If the data file is ahead (it is assumed that there is missing data) that station is not processed until data is available. Either RDNGS or RDRNX is called to read records from the input file depending on the type of file which is determined in CHKTYPE. INITDT is called before the records are read, and LOADDT is called once the data to be processed has been read in.

CHKTYPE: Open and check the type of the ***.DAT** *files*

Each input ***.DAT** file is opened and the first line is read. The file type, either NGS or RINEX, is determined and saved. The station name is read from the file. The files are left opened, but the RINEX files are rewound so that the headers may be searched again.

CLKPOL: Solve for station clock polynomial coefficients

The normal matrix for each station (computed in PEDIT) is solved to determine the station clock polynomial coefficients. The matrix is solved using SIMEQ.

ELAVE: Find Average Keplerian Elements

The average Keplerian elements for the observation period are calculated by converting the earth-fixed coordinates given in the precise orbit file to the inertial frame (using SIDTM and RINRL) computing the Keplerian elements (using ELXYZ) and averaging for all records in the observation period for each satellite used.

ELXYZ: Calculate the Osculating Orbital (Keplerian) Elements of a satellite

Given the inertial coordinates of a satellite ELXYZ computes the osculating orbital elements. This is done using well known geometric formulas (Bate, 1971, Escobal, 1965).

GETDAT: Read and process data epoch-by-epoch

This is the core of Merge. This routine corrects for offsets from master time, combines data from different stations, and writes the **<DB>DT.DAT** and the **<DB>OR.DAT** files. The main loop for the entire program is contained in GETDAT, and is described in the following sections.

Open Datbase Files

The **<db>DT.DAT** and **<db>OR.DAT** files which will be written to by WRDDAT at the end of the main loop are opened.

Eliminate SVs from requested list if no orbit data found

The list of SVs supplied by the setup file is checked against the data read from the broadcast ephemeris file. Any SVs not found are removed from the list. This is repeated for the precise orbit file if one is being used. The dropped SVs are displayed on the screen.

Initialize the counters for the main loop

There are two counters used. One for the number of records written to database file, and one for epochs on the master clock. These will get out of sync if no data is recorded for some epochs. The start and stop times are set in terms of fractional days of year.

Read RINEX header records

For all input files that are RINEX type the file header is read using RNXHDR.

Step through data

Start the main loop. The master time is set to the start time input parameter and incremented, and the loop is ended when the end time is reached. The main loop is also ended if there are multiple receivers (the normal situation) and the active station count drops below two.

CHKSEQ is called to read the record headers and check the time against the master clock.

CHKORB is called to check to see if new orbit polynomials are needed. Orbit polynomials are updated every 3.5 hours. The actual time check is done in a call to CHKORB. At this point if something has gone wrong and the master time is earlier than the time period that the current orbits polynomials are valid the main loop is stopped. If there is no data for the current epoch then the rest of the main loop is skipped. Otherwise the first receiver with data is chosen as a reference and CHKORB is called.

POSVEL is called to compute satellite positions and velocities based on the receiver clocks. MOBILE is called to update the position of the mobile stations if merge computing a kinematic solution. POSVL2 is called to compute satellite positions and velocities based on the master clock. ADJDAT is called to adjust all phases and ranges using master clock. PEDIT is called to edit out the large cycle slips in the data. TDOBS is called to load the triple difference observation equations. WRTDAT is called to output data to the **<db>DT.DAT** and **<db>OR.DAT** files

This is the end of the main loop. If the loop ended because there was an error with the orbit polynomials, an error message is displayed and written to the summary file.

Return to main program merge

*GETORB: Open the broadcast orbit file and the **BCSEL.OUT** file*

GETORB opens the ***.ORB** file specified by the input parameter file and calls BCREAD to process it. If precise orbit file is being used, a scratch file is opened for rapid direct-access and ARCFIT is called to process the precise orbit file.

INITDT: Initialize arrays

Input data arrays for the given station/satellite pair are initialized.

INITLZ: Initialize all variables

Symbolic constants for Logical unit numbers are set. All arrays used to accumulate data are initialized to 0. Some flag arrays are initialized to -9999.0. This value is used throughout XOMNI to indicate that no data is present. Arrays used to hold minimums and maximums are initialized to large positive and large negative numbers respectively.

LOADDT: Load Data Arrays

LOADDT is called for each station once per epoch. It loops over each satellite with input data available for that epoch. If the satellite is available but not selected in the parameter file a message is displayed on the screen and stored in the summary file. A list is maintained of all satellite observed by any receiver in the current epoch. The data read from the input files earlier is stored in the appropriate arrays. The data valid flag is set based on the input data. Station clock corrections are applied to the pseudo-range or phase data, or both, or neither depending on the LSVCLK parameter.

MOBILE: Update position of moving receivers

For each station with an ISLV parameter equal to 2 (MBL) a position for the current epoch is determined using the pseudo-range by calling RANGE, and phase residuals computed by calling RESID.

PEDIT: Edit large cycle slips, write various plot files

The double difference doppler velocity residuals are formed and written to the PXA plot file. The elevation data is written to the ELV plot file. The pseudo-range clock error/range residual is computed by subtracting the computed satellite-receiver distance from the raw pseudo-range. The pseudo-range clock error is written to the CLK plot file. The range residual normal matrix is loaded with the value for the current epoch. The matrix will be used to solve for the residual clock error as a second degree polynomial with respect to time.

The phase data is then checked for large cycle slips if the input parameter IEDIT is non-zero. This is done by comparing the current phase residual with the two previous phase residuals. This check is only done if two previous observations are available. The time between the first and last observations to be considered must be less than 1 hour, and the change in phase between consecutive observations is within a limit. If the change in slope between the two previous points and the current point must also be within a given limit. If these conditions are not met the current station/satellite/frequency combination is skipped until they are. The check for a cycle slip is then performed by extrapolating what the current phase residual should be based on the previous two values. If the actual observed value is off by to much a correction is computed and added to the data from then on.

PLTLIM: Write plot limit files

This subroutine generates the plot limit files used by the plotting program for the initial limits of the plots. The plot limits for each plot have been previously calculated by the routines generating the plots. For each plot the limit filename is generated, the file is opened, the start and stop times are written, and the minimum and maximum values for each satellite/station combination are written out.

POSVEL: Compute satellite positions and velocities at station clock times

For each active station the position and velocity of each satellite in the current epoch is computed using the precise orbit data if it is available, or the broadcast data. In the case of precise orbit data the computation is a simple polynomial expansion based on the difference between the station time and the current orbit start time (computed in GETORB and CHKORB when new polynomial coefficients are computed by ARCFIT). When using precise orbit data the satellite acceleration is also computed. In the case of broadcast orbit data BCORB is called to compute the satellite position and velocity (no acceleration is computed). RESID is then called to compute the phase residuals for each satellite.

POSVL2: Compute satellite positions and velocities at master clock time

POSVL2 works identically to POSVEL except that only the master clock is used as a reference time to compute satellite information and RESID is not called. Because a common time is used the satellite positions are recomputed only once, not for each station.

RANGE: Calculate position of mobile stations

RANGE is called once for each mobile station. If there is only one mobile station then an MBL plot and a KINSLV file are generated. If these files are being generated then the first time RANGE is called the MBL plot file is opened and the header is written and the KINSLV file is opened. The pseudo-range residuals are formed based on the input positions of the mobile station and the reference station. A normal matrix is formed and if there are enough observations (4) SIMEQ is called to solve for the error in the input position of the mobile station. If a valid solution is found then the mobile station's position is updated, the north/east displacements from the start position are computed and written to the MBL plot file. VELSLV is then called to compute the mobile station velocity.

RDINP: Read the MERGE setup file

Setup file, **MERGE.INP**, is opened, parameters are read, and the file is closed. See appendix for input parameters.

RDNGS: Read NGS format data records

RDNGS reads one record from an NGS format ***.DAT** file. First the time tag is read along with the number of satellites and a list of satellite PRNs. For each satellite the phase and pseudo-range data is read. If the end of file is reached then that station is marked as inactive.

RDPOM: Read position, offset, and meteorological data

The ***.POM** files are opened and read. These files contain position, offset, and meteorological data. The monument positions are converted to antenna positions using the offset data. The sation names are displayed on the screen along with the other permanent parts of the status display

RDRNX: Read RINEX input data file

RDRNX reads one record from a RINEX data file. Only a subset of the RINEX standard (Gurtner, 1990) is implemented, but it covers what normal GPS receiver data will include. First the time-tag and satellite PRN list is read. The time from the file is converted to GPS seconds of week using IGPSWK. Each data item is then read and stored into the appropriate array. There are three versions of RDRNX, one for each version of merge. These versions differ in there handling of the pseudo-range data. For the "c" version of merge, currently "merge320c," only the C/A code pseudo-range is considered, the P code is ignored. In the "cp" version the P code is used if available otherwise the C/A code is used. In the "p" version only the P code is used, and the C/A code is ignored.

RESID: Compute phase residuals

RESID is called for each station/satellite pair. The distance between the station and satellite is computed based on the current station/satellite positions. The satellite position is then adjusted for the time the signal took to reach the station. Radial velocity and radial acceleration of the satellite is computed. Elevation angle of the satellite is computed. This is checked against the

elevation cutoff parameter (ELVCT) and if it is less the record is marked as bad. The tropospheric correction is then computed. A theoretical phase is computed based on the distance to the satellite and the tropospheric correction. The phase residual is computed as the difference between the theoretical phase and the measured phase.

RNXHDR: Read RINEX header

RNXHDR reads the header of a RINEX (Gurtner, 1990) data file. Most of the data is ignored. Only the start time, number of data items, and data type sections are used by MERGE.

TDOBS: Form double and triple difference phase observables

For each station/satellite pair the double and triple differences are formed. First a reference satellite is picked simply by choosing the first available satellite. The data validity is checked for each combination of reference and non-reference satellite and reference and non-reference station. If all the data is valid the phase residual double difference is formed. This is then subtracted from the previous double difference to form the triple difference. The normal vector is formed and if there is no apparent cycle slip between this and the last observation it is added to the normal matrix. The triple difference phase residual is written to the TDF plot file.

TDSOLV: Do a triple difference solution for the non-reference stations

Now the triple difference solution is actually computed. This is relatively simple because the observation matrix has already been created by TDOBS. The matrix is solved using SIMEQ and displayed on the screen and written to the summary file.

TDSOLV: Solve and output triple difference solution

TDSOLV first calls SIMEQ to solve the triple difference solutions accumulated by TDOBS. If there is an error generating the solution a message is output to the summary file and displayed on the screen. In any case the solution is then output to the screen and summary file. The baselines between all the solved and the reference station are output to the summary file.

VELSLV: Solve for mobile station velocity

VELSLV generates a normal matrix using double difference doppler velocity data. The reference satellite is chosen as the first found with good doppler data. The normal matrix is formed and SIMEQ is called to solve for the station velocity.

WRTDAT: Write data to output database files

The current solution time is written to the screen and the summary file. The satellite PRN list and the current time tag is written to the DT output file and the OR output file. The doppler velocity data is written to the AX output file. The current satellite positions and velocities are written to the OR file. The phase and pseudo-range data, edit information, and tropospheric correction for each station/satellite pair are written to the DT file.

WRTHD: Write database header file

WRTHD first calls ELAVE to compute the average orbital elements for each satellite. The following information is then written to the database header file:

- Start and stop time
- Number of stations and satellites
- Satellite PRN and average orbital elements Position and Meteorological data for each station

GPS22

Main program

Figure 2 shows the overall execution flow for GPS22.

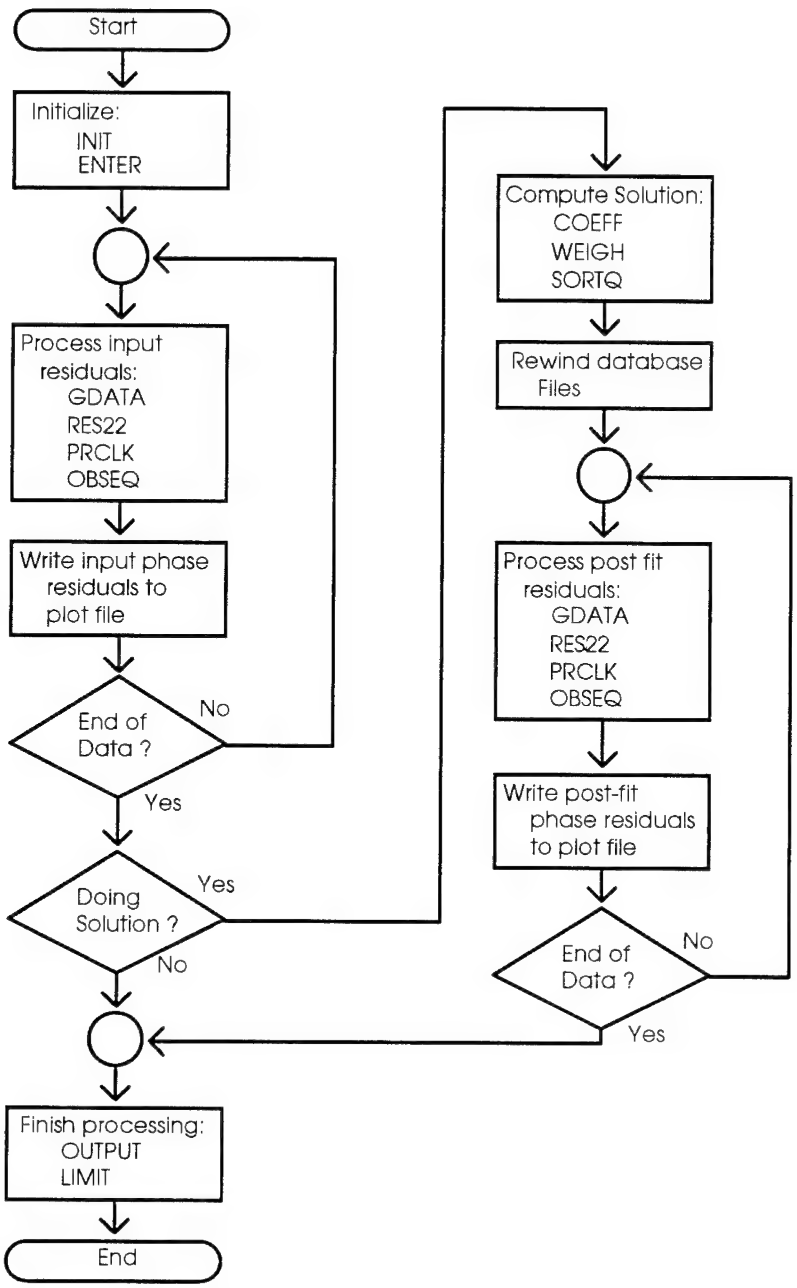

Figure 2 – GPS22 Flowchart

Initialize Parameters and Get User Inputs

INIT is called to initialize global variables and constants. ENTER is called to read the setup file.

Start Main Loop Over Data for Solution

This first loop is used to form input double difference phase residuals. If the processing mode is A PRIORI (LSOLN=0) then processing stops after this loop ends, otherwise a solution is generated and another pass over the data is made to compute post-fit residuals. An appropriate message is displayed for each pass, depending on the processing mode.

GDATA is called to read a record from the input database. If either the stop time or the end of the database file is reached then loop is terminated. If processing continues RES22 is called to form the input residuals. PRCLK is then called to compute a clock model. OBSEQ is called to compute partial derivatives and the observation equation.

If this is the first observation the database record time is stored in case it is different from the user input start time. The start time is then written to the **res.dat** file. For all records the number of satellites in the current record and the reference satellite PRN are written to the **res.dat** file. For each non-reference satellite the single difference is formed and written to the **res.dat** file. The double difference phase residuals are written to the plot file, and minimum and maximum values of are tracked for use in the limit file. Control now returns to the beginning of the main loop.

Generate Solution

As mentioned earlier if the processing mode is A PRIORI then this section and the following one are skipped. If the processing mode is SOLUTION then COEFF is called to write the normal matrix and other internal values for examination, WEIGH is called to apply weights to station positions, and SORTQ is called to generate the solution.

Loop over data to compute post-fit residuals

The input database is re-read, and the post-fit residuals formed by calls to GDATA, RES22, PRCLK, and OBSEQ. The residuals are written to the plot file and the minimum and maximum values stored. The sigma value computed by OBSEQ is normalized based on the number of observations used and the number of parameters estimated.

Print results

If the processing mode is SOLUTION the generated solution is written to the **SAVIT** file. All output files are closed. LIMIT is called to generate the limit files for the plots. The program is then terminated.

INIT: Initialize Global Arrays and Constants

This routine sets all global variables to their proper values.

ENTER: Process Setup and Database Header Files

Open output files

The **SAVIT**, **NRMTX**, **res.dat**, and **DUMP** files are opened.

Read setup file

Various parameters are read from the setup file, **GPS22.INP** (see appendix for details). Input parameters are written to the summary file, **GPS22.SUM**, in a human readable format.

Read database header file

The station and satellite information is read from the database header file, **<db>HD.DAT** (see appendix for details).

Compute Satellite Sequence Array

An array is computed to map satellite PRN to non-omitted satellite index. Read integer file. The integer file, **INTGR**, is read if it exists. This file contains the biases calculated in a previous run of GPS22.

Process Station data

The following is done for each station:

Input X,Y,Z coordinates are converted to latitude and longitude. Using input parameter LSTA, the reference station and fiduciary stations are identified and an array mapping stations to

be solved for to their indexes is computed. Using input parameter LCLK an array mapping stations which require a clock solution to their indexes is computed. Using input parameter LHGTS an array mapping stations set for height solution to there indexes is computed.

Open plot file

All plot files are opened and the headers are written.

Write setup summary

Total number of parameters to be solved for is computed. Information on type of parameters to be solved for is written to the summary file. If the total number of parameters exceeds the limit of 200 then a message is displayed and the program is terminated.

COEFF: Print Correlation Coefficients

Loop over each station and print out correlation coefficients as described in the OMNI documentation.

GDATA: Read database files

The current record is read from the **OR** and **DT** files. If the current time is before the start time records are read until it isn't. If the time is after the end time or and end-of-file condition is detected a flag is set indicating the end of data and the procedure returns.

If only one reference satellite is specified in the parameter file, it is checked. If it is available for the current epoch and has valid data, it is used; otherwise, the first valid satellite found is used.

If a list of reference satellites is specified, the one specified for the current epoch is used *without* checking for validity.

The edits stored in the database are applied to the phase data. The current reference satellite is written to the summary file if it changed since the previous epoch. For each satellite not seen before during this processing run, RINRL and ELXYZ are called to compute the orbital elements.

LIMIT: Write limit files

Start/stop times and min/max values are written to the ***.LIM** files, one for each ***.PLT** file.

OBSEQ: Compute partial derivatives and observation equation

All phase residuals are adjusted using the time offsets computed in PRCLK to be referenced to the same time. RPART is then called to compute the partial derivatives of each satellites position in inertial space (which is computed with RINRL) with respect to its orbital arc elements. The partials are then converted back to earth centered-earth fixed coordinates by RECEF. If the solution has already been computed then the satellite position is updated using the partials and the corrections to the elements that were computed.

For each satellite/station pair the normal matrix is loaded for each parameter to be computed. These are: the clock terms, the satellite biases, the satellite orbit arc elements (if selected), the tropospheric scale height, and the unknown station positions.

OUTPT: Generate output summary file

COEFF is called to print the correlation coefficients. Tropospheric correction summary is printed. For each station a solution summary is printed. The solution summary includes: original and corrected position, offset in XYZ and lat/lon/height coordinates, corretion sigma, and bias terms. The satellite arc adjustments are then printed if they were selected.

PRCLK: Compute clock error from pseudo-range

The station-satellite time offsets are computed based on the input station positions and the current pseudorange values. The time offset for each station is then computed either by averaging the station-satellite offsets (if LCLK is 0 for the station) or from the clock polynomial coefficients read from the database header file (if LCLK is 1).

RECEF: Rotate from inertial space to the earth fixed space

This is a simple rotation based on siderel time which is passed in as a parameter.

RES22: Compute phase residuals

First the earth tide correction (computed with XTIDE) and satellite center of mass correction are applied to compute the station to satellite distance for each satellite/station pair. Radial velocity and elevation angles are computed. The troposphere correction is then applied if LTROP

is 1. The theoretical phase is then computed from the station to satellite distance. The phase residual is computed by taking the difference between the recorded phase and the theoretical phase.

RMATX: Rotate errors from XYZ to North/East/Up

This is a simple rotation and assumes a spherical earth.

RPART: Compute partial derivatives of satellite position

The partial derivatives of the supplied satellite position vector with respect to the orbital elements are computed. These equations are well known (Hofmann-Wellenhof, et al., 1993).

SORTQ: Identify valid portions of normal matrix and solve

The normal (or solution) matrix contains entries for all solveable parameters. For any given run depending on flag settings, satellite availability, and station setup not all of these parameters will be active. SORTQ determines the empty rows and columns of the normal matrix and compresses it before calling SLV22 to compute the solution. After the solution is computed the empty rows and columns are re-inserted.

TIDPOT: Tide-generating potential calculation

This subroutine is an implementation of the tide-generating potential as given by (Cartwright and Tayler, 1971, Cartwright and Edden, 1973).

WEIGH: Apply constraints to satellite orbit adjustment

The weighting in the normal matrix is increased for stations that are identified as "Fiduciary" by the LSTA parameter.

XTIDE: Calculate solid earth tide displacements

Using TIDPOT, 3 polynomials are calculated for each station estimating the X, Y, and Z displacements due to earth tide potential. These polynomials are then evaluated to generate the

displacements for each epoch. If two hours have elapsed since the last polynomial generation the coefficients are recomputed.

XYZNEU: Convert a vector from XYZ to NEU

Simple coordinate conversion from X, Y, Z to North, East, Up. Note that this routine is for relative positions (vectors) not absolute positions.

NAV22

NAV22 is the navigation processor for XOMNI. As with GPS22, NAV22 requires a merge database and a setup file as input. It also can optionally use several other files for input. These files are the bias and jump files, which allow greater control over the ambiguities used by NAV22.

The output that NAV22 produces depends on the mode it is run in. The mode is selected by in the setup file. The two main modes are "SOLUTION" and "A PRIORI." In "SOLUTION" mode the output includes a navigation file (the main output), several plot files, and a summary file. In "A PRIORI" mode the output includes an edit file along with some different plot files. There is a third mode available, called "PREPROCESS." This mode is experimental and should not be used.

NAV22 requires that the position in the database header file is as precise as possible. This position is used to compute the initial ambiguities, and all other positions are based on this. Normally a static solution is produced with GPS22 that is used to update the position in the header file.

Input files:

Filename	Described In	Mode
NAV22.INP	NAV22/NAV22	both
bias	NAV22/NAV22	solution
jump	NAV22/PHASE	solution
<db>DT.DAT	MERGE/WRTDAT	both
<db>HD.DAT	MERGE/WRTDAT	both
<db>OR.DAT	MERGE/WRTDAT	both
<db>AX.DAT	MERGE/WRTDAT	both
ANTSWAP.OUT	NAV22/NAV22	both

Output files:

Filename	Described in	Solution/A Priori
NAV22.OUT	NAV22/NAV22	both
NAV22.SUM	NAV22/NAV22	both
NAV22.EDT	NAV22/CYCHK	a priori

Main program

Figure 3 shows to overall execution flow for NAV22.

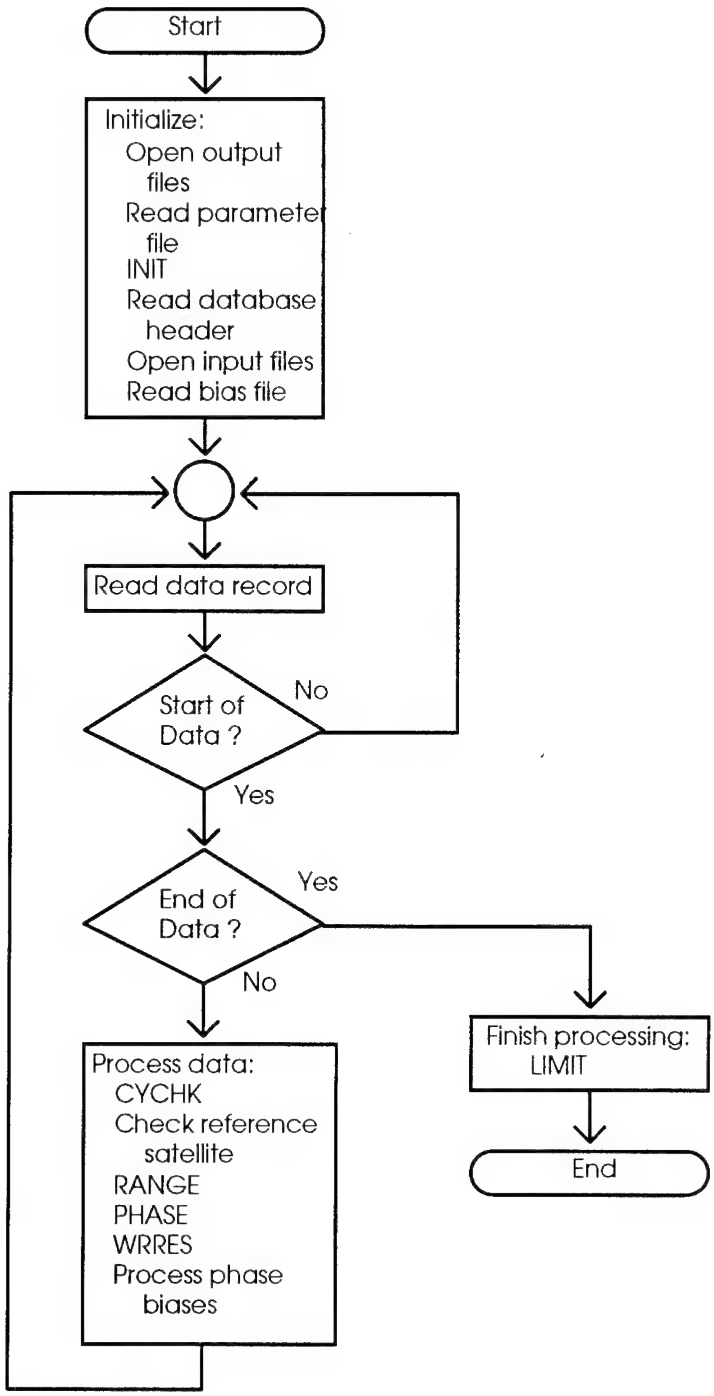

Figure 3 – NAV22 Flowchart

The purpose of the main program is to open all input and output files; read and process the setup files; perform initializations; and process the main loop.

Read and process setup files

The NAV22.INP file is read to determine various flags, start and stop time, reference satellite scenario, and station information. At this point if the antenna swap flag is on the ANTSWAP.OUT file is read. Antenna swap is not supported by XOMNI, so this flag should be left off.

Perform initializations

Antenna position is computed based on benchmark position and antenna offset. **HEADS** is called to write output file headers.

Process main loop

In the main loop the <db>DT.DAT and the <db>OR.DAT files are read one record at a time. The following processing is performed on each record.

Check time

The time from the data and orbit files are checked and the program is terminated if there is a discrepancy (this will only occur if the input files have been corrupted) . The time is then checked against the start and stop times given in the setup file. When in "a priori" mode the start and stop times are ignored, and the entire file is processed. Otherwise, records are read from the input files until the start time is reached, and processing stops when the end time is reached. When the start time is later than the stop time the files are read backwards and processed in reverse.

Check enough data

Now the data is checked to determine whether there are enough observations in the current epoch to compute a solution. If fewer than the required 3 double differences can be created the next record is read and a message is displayed and written to the output file. Again, if NAV22 is in "a priori" mode then this check is skipped and all data is processed.

Apply corrections

The adjustments stored in the input data file are added to the phase data values for valid elements.

Display message

Every 50 records the current record and time are displayed.

Check slips

CYCHK is called to check for cycle slips if in "a priori" mode.

Check/change reference

If a reference scenario exists this is checked first to determine if the reference satellite should be changed. If this is not true then the current reference is checked to make sure valid data exists. If no valid data exists then the other satellites are searched to check for a possible new reference.

If a new reference satellite has been chosen in one of the two above ways then the following additional processing occurs: The old and new reference satellites are logged to the output file. The phase ambiguity of the new reference is subtracted from phase ambiguities of all the other satellites (this simplifies the phase solution calculation that occurs later in the processing). The process is then repeated to make sure that the new reference has valid data.

Compute range and phase solutions

RANGE is called to compute a pseudo-range solution if selected. Data from the bias file is checked to see if any ambiguities need to be reset. PHASE is called to compute a phase solution. If NAV22 is in "preprocess" mode then WRRES is called to write the phase differences to the phase output file.

Fix ambiguities if required

If LBIAS is set to other than float (0) then the ambiguities are either rounded to the nearest whole cycle (half cycle when appropriate) or if it is the first time a satellite has appeared and there is an entry in the **IBIAS** file for it then they are set to that value.

At this point the main loop repeats.

CYCHK: Check for cycle slips

Compute and Write Ionosphere Residuals

The phase ionosphere residual is formed for all satellite/station pairs and written to the plot file. Minimum and maximum values are tracked to be written to the limit file.

Compute and Write Range Residuals

The range residual is formed for all satellite/station pairs and written to the plot file. This is done for both frequencies if data is available. Minimum and maximum values are tracked to be written to the limit file.

Check for Cycle Slips

If a slip is detected in the ionosphere residual and the processing mode is SOLUTION mode or the selected frequency is a combined frequency, L3 or L3AVE, then an edit is written to the NAV22.EDT file. If a single frequency mode has been selected then the range residuals are check for cycle slips. If any slips are detected in either residual a message is logged to NAV22.OUT.

DPHI2: Compute Phase Residuals

This routine loops through each satellite and each station for all of the following computations:

Compute Distance from Satellite to Station

Using the current positions the straight line distance between the current satellite and stations is computed.

Correct For Antenna to Center of Mass Displacement

The satellite position stored in the database is referenced to the satellites center of mass. This is adusted to the positon of the satellite antenna

Apply Tropospheric Correction

The elevation angle from the station to the satellite is computed. TLATE is called to convert the current station position to lat-lon-hight. STROP is then called to compute the troposphere correction.

Compute the Radial Velocity

Radial velocity is computed from the XYZ velocity read from the **.ORB** file.

Compute The Theoretical and Residual Phase

The theoretical phase is computed using the distance from the station to satellite (using the current assumed positions) and the troposphere correction. The phase residual is the difference between the measured phase and the theoretical phase.

HEADS: Write Headers to All Plot Files

Each plot file is opened and a header is written describing the data to be written.

INIT: Initialize Global Arrays and Constants

Global arrays and various constants are initialized.

LIMIT: Write Limit Files

For each plot file there is a limit, ***<doy>.LIM**, file. These files contain the start and stop times for the plot files and the minimum and maximum values for each set of data in the file. They are opened, written to, and closed by LIMIT.

PHASE: Generate Phase Solution for Current Epoch

This routine is the heart of NAV22. The double difference phase residuals are computed and formed into a normal matrix. SLV22 is called to find the least squares fit of the error in the current position. The position is adjusted and the process is repeated.

Update Position using Doppler Data

If the RDOP is good enough then the current position is updated using the doppler data. By doing this the input phase residuals can be examined to detect cycle slips (this is done later in PHASE).

Compute Theoretical Phase Residuals

DPHI2 is called to compute the phase residuals.

Compute Double Differences

For each non-reference satellite with valid data the phase residual double differences are formed.

Compute Smoothed Ionosphere Correction

If the selected frequency is L3AVE the ionosphere delay is computed separately and a moving average is applied to the phase data. The computed ionosphere delays are stored in an array. This array is shifted, the new value is inserted, and the average is computed. This value is then written to a plot file, and the minimum and maximum values are tracked.

Process Jump File

The **jump** file contains time records where the phase data is known to have errors. If the current time matches the next time in the jump file, and it was possible to update the current position with the doppler data, then the position is not updated using the phase residuals and the phase biases for all satellites are reset based on the current position. The next iteration of the phase solution is skipped and processing jumps to the section below labeled "Compute Bias Terms."

Check Input Phase Residuals for Cycle Slips

If this is the first iteration, and the current position was updated with the doppler data, and the reference satellite didn't change then the input phase residuals are checked for cycle slips. If a slip is found a corresponding edit is written to the **NAV22.EDT** file. A message is also written to the **NAV22.OUT** file. If a single bad point is detached a message is only written to **NAV22.OUT**.

Define Normal Matrix

The normal matrix is computed from the double difference phase residuals and normal vectors based on the current mobile station position.

Solve Normal Matrix

SLV22 is called to solve the normal matrix equation. It is actually called twice: once to reduce the matrix and again to compute the inverse.

Update Mobile Station Position

The current mobile station position is updated using the results of the normal matrix solution which include the position error. The solution is then iterated once using this new position by jumping back up to the section labeled "Compute Theoretical Phase Residuals."

After the second iteration, the mobile station velocity is computed using the position from the previous epoch. The position and velocity information is then written to the **NAV22.OUT** file.

Compute Bias Terms

At this point, the position has been updated either by using the Doppler data or by iterating the solution using phase residuals. The bias terms for any new satellites and any satellites reset because of either the **jump** or **bias** files are computed based on this position. The final double difference phase residuals are the written out to the PRES plot file.

RANGE: Compute Pseudo-range Solution

Check for range solution

If a pseudo-range solution is not being done, i.e. LRNG is 0, the station time offsets are computed based on the clock polynomial coefficients read from the database header file. RANGE then returns.

Reload starting position

If all plot mode has been selected, i.e. LPLOT is 2, then the original input position for the mobile station is reloaded as the initial guess for a solution, otherwise the previously computed position is used.

Compute Range Residuals

DPHI2 is called to compute theoretical range based on the current guess for a solution.

Range residuals are formed by subtracting the theoretical range from the pseudo-range. If a combined frequency solution is being performed (LRFRQ equals 3 or 4) then a combined residual is formed.

Estimate Average Reference Clock and Residual SV Clock Errors

The Average clock error is estimated by averaging the pseudo-range residuals. The Residual Clock error is computed by subtracting this average from each pseudo-range residual.

Define Normal Matrix for Mobile Station

Normal vectors are computed based on current receiver positions. The Normal matrix is constructed from these vectors and either the clock corrected pseudo-ranges (LRNG=1) or the differential pseudo-ranges (LRNG=2).

Write Pseudo-range Data to Plot File

If All Plot mode is selected (LPLOT=2) the raw pseudo-range residuals are written to a plot file.

Solve Normal Matrix

If sufficient observation data exists (4 or more SVs observed) SIMEQ is called to solve for the offset to a new position. If a singularity is detected a message is displayed and the program is stopped.

Adjust Current Position

The computed offset is added to the assumed position to form the new pseudo-range estimated position. North, East and up displacements from the starting position as well as velocity in the X, Y and Z directions are computed and are written, along with the current XYZ position, to the range position plot file.

STROP: Compute Troposphere Delay

Compute troposphere delay based on temperature, humidity, elevation angle, atmospheric pressure, latitude, and altitude of receiver.

WRRES: Write Raw Residuals

This routine is only called when NAV22 is in "preprocess" mode (LSOLN=3). It recomputes the double difference phase residuals and writes them out along with the double difference pseudo-range and the final position computed for each epoch.

Common Routines

These routines are used in two or more of the programs MERGE, GPS22, and NAV22.

MDHM: Convert Day of Year to Month and Day of Month

The algorithm used is a simple table look up. It is valid until 2100 which is well beyond the expected lifetime of this software. Total seconds of day is also converted to hours, minutes and seconds.

MJDAY: Convert Year, Month, Day, to Julian Days since 1900

Again, a simple table lookup valid until 2100.

NEUXYZ: Convert a vector from NEU to XYZ

Simple coordinate conversion from North, East, Up to X,Y,Z. Note that this routine is for relative positions (vectors) not absolute positions.

RINRL: Convert from Earth-Centered-Earth-Fixed to Inertial Coordinates

Using the input sidereal time the earth-centered-earth-fixed position and velocity are converted to inertial coordinates. This is a simple computation once sidereal time is known (Aoki, 1982, Mueller, 1977, Kaula, 1966).

SIDTM: Convert Calendar Date and Time to Sidereal time.

The year is converted to years since 1900. The time, passed as hours minutes and seconds based on GPS time, is converted to UTC seconds. This is then converted to Julian days since 1900. Sidereal time is then computed based on a second degree approximation. The time is returned as decimal hours(Robertson, 1975).

SIMEQ: Solve System of Linear Equations.

The set of equations is passed in an augmented array and is solved by Gaussian elimination and back-substitution. A flag is returned to indicate the number of singularities found (based on the input tolerance value) when attempting to solve the system (Faddeeva, 1959).

SLV22: Reduce, Solve and Invert Symmetric Normal Matrices

This routine can be called in three different ways. The first mode uses Choleskey factorization to reduce the input matrix. The second mode uses the reduced matrix to compute a solution given one of more input vectors. The third mode computes the full inverse matrix from the reduced matrix.

TLATE: Convert Between Geodetic and Cartesian Coordinates

This routine will convert either from geodetic to Cartesian or Cartesian to geodetic depending on input flags. The conversion from geodetic to Cartesian is a straightforward computation. The reference ellipsoid parameters (semi-major axis and flattening) are passed in as parameters. The conversion from Cartesian to geodetic is computed through an iterative approximation (Soler and Hothem, 1988, Rapp, 1984).

FUDGE

FUDGE operates on a file, **res.dat**, produced by gps22e and produces edit, fudge, and bias files. There are two main differences between FUDGE and earlier cycle slip correcting procedures used with XOMNI. The first is that FUDGE does not operate on double differences. In the previous editing procedures double differences where formed based on a reference satellite and a reference receiver. These double differences were then examined for cycle slips. If there was no data for the reference receiver then that observation had to be skipped. FUDGE, on the other hand, takes as input single differences based on a reference satellite and forms all possible double differences. There is no single reference receiver and so when a data gap occurs only the receiver with the gap is affected. This also allows better checking in the presence of noise because more double differences are looked at. For example, with 4 receivers and a single reference you get 3 double differences. FUDGE looks at all 6 possible double differences. The second difference between FUDGE and earlier editing techniques is that FUDGE provides a way to fill in gaps in one receiver using data from others. Data for a receiver can be reconstructed from data from another because in the absence of cycle slips and noise double differences are constant if the receiver positions are known.

MAIN

GET_ARGS is called to check command line arguments. GET_INCREMENT is called to read MERGE.INP file. READ_HEADER is called to read input file header. INIT is called to initialize arrays. MAIN then loops over data calling READ_PRES for each observation. The loop ends when READ_PRES returns 0. CHECK_JUMPS, FIX_JUMP, and UPDATE are called for each observation. CLEANUP is then called to generate any final edits required.

GET_ARGS

Check for options. Supported options are -d which turns on debug mode, -i to manually set increment, and -2 to indicate that L2 data is full cycle.

The default input file name is **res.dat**. If an argument is supplied on the command line it is taken as an alternate input file name. Input file is opened.

If there is an error reading the options, opening the input file, or extra arguments are supplied then usage is called. Output files are opened: g.edt, n.edt, bias, log

GET_INCREMENT

If increment was not set on the command line then attempt to open file **MERGE.INP** for reading. If this is successful the increment is read. If there is an error opening the file or reading increment then **usage** is called to indicate that the increment must be set on the command line.

READ_HEADER

The header of the input file is read. This consists of a format version, number of receivers, and start time (DOY, hour, minute, second). The format version is checked for validity, if it is bad then an error message is displayed, and the program exits. If the format version is greater the 1.00 then the FUDGE operation is possible and an extra output file, **n.fdg**, is generated.

The start time is converted to the internal format which is seconds since the start of the day.

If the Debug option was selected the contents of the file header are printed to the log file.

INIT

An array is initialized that stores strings describing the various status values assigned to differences.

APPENDIX A

PARAMETER FILE DESCRIPTIONS

Each of the three main XOMNI modules reads an input parameter file. These files are normally created by the setup programs as described in the User's Guide. However, it is sometimes desirable to edit these files manually. In the following file descriptions each field in these files is represented by a string of repeated characters. The number of characters represents the number of spaces alloted to each field. This spacing must be maintained for correct processing of these files.

Input file for merge: MERGE.INP

```
    0
    2   11  2  7  9 12 13 15 19 24 26 27 31
       1.00      12.00    1    0
  216   13    5   .00  216   17    6   .00
a216
aaf3216a
    0    0    3
flt1216a
    0    2    3
aaf3216a
    1
ngs07083
```

Fig. 4 – Example **MERGE.INP** file

```
AAAAA
BBBBBCCCCCDDD...
EEEEEEE.EEFFFFFFF.FFGGGGGHHHHH
IIIIIJJJJJKKKKKLLL.LLMMMMMNNNNNOOOOOPPP.PP
QQQQ
RRRRRRRR
SSSSSTTTTTUUUUU
...
VVVVVVVV
WWWWW
XXXXXXXX
```

Fig. 5 – Fields for **MERGE.INP** file

Table 1 – **MERGE.INP** Field Descriptions

Field	ID	Format	Description
A	LQUIT	(I5)	Flag indicating whether to run merge (should always be 0).
B	NSTA	(I5)	Number of stations.
C	KSAT	(I5)	Number of satellites.
D	IDSV	(I3)	Satellite PRNs. Repeated KSAT times.
E	TINCS	(F10.2)	Time increment in seconds.
F	ELVCT	(F10.2)	Elevation cutoff angle, in degrees.
G	LFRQ	(I5)	Frequency used for triple difference solutions. 1=L1, 3=L3
H	IMCLK	(I5)	Master clock option. If not 0 it indicates which station to base the master clock on.
I-L		(3I5,F5.2)	Start time. Fields are: DOY, Hour, Minute, Second.
M-P		(3I5,F5.2)	Stop time. Fields are: DOY, Hour, Minute, Second.
Q	NOUT	(A4)	Database ID for output files.
R	NIN	(A8)	Filename base for station input file. Extensions will be added. This and next 3 fields are repeated for each of NSTA stations.
S	IEDIT	(I5)	Flag indicating whether to do automatic cycle slip editing. 1=YES, 2=NO.
T	ISLV	(I5)	Solution Status. -1=OMIT, 0=REF, 1=SLV, 2=MBL.
U	LSVCLK	(I5)	Clock correction flag. 0=NONE, 1=RNG, 2=PHASE, 3=BOTH.
V	NUORB	(A8)	Broadcast orbit filename base. Extension ".ORB" will be added to form filename.
W	LORB	(I5)	Flag indicating orbit type. 1=PRECISE, 2=BROADCAST. A broadcast orbit file is required with either option.
X	NEPH	(A8)	Precise orbit filename base. Extension **.EPH** will be added to form filename. This field is only read if LORB is 1.

Input file for gps22: GPS22.INP

```
         0
a180
         0      0
         0      0      0     .0   365     24      0     .0
         1      1      0
    0
         0
        13
         4
tul1                     1     -1      0
tul2                     0     -1      0
tul3                     1     -1      0
tul4                     1     -1      0
```

Fig. 6 – Example **GPS22.INP** file with single reference satellite

```
AAAAA
BBBB
CCCCCDDDDD
EEEEEFFFFFGGGGGHHH.HIIIIIJJJJJKKKKKLLL.L
MMMMMNNNNNOOOOOO
PPPQQQ...
RRRRR
SSSSS
TTTTT
UUUUUUUUUUUUUVVVVVWWWWWXXXXX
...
```

Fig. 7 – Fields for **GPS22.INP** file with single reference satellite

```
      0
 a180
      0     0
      0     0     0    .0   365    24     0    .0
      1     1     0
   0
      0
      0
     13   180     0     0     0
     24   180    22    30     0
      3   181     1     0     0
     17   181     2    30     0
     23   181     4     0     0
      0
      4
 tul1               1    -1     0
 tul2               0    -1     0
 tul3               1    -1     0
 tul4               1    -1     0
```

Fig. 8 – Example **GPS22.INP** file with multiple reference satellites

```
AAAAA
BBBB
CCCCCDDDDD
EEEEEFFFFFGGGGGHHH.HIIIIIJJJJJKKKKKLLL.L
MMMMMNNNNNOOOOO
PPPQQ...
RRRRR
SSSSS
1111122222333334444455555
...
66666
TTTTT
UUUUUUUUUUUUUVVVVVWWWWWXXXXX
...
```

Fig. 6 – Fields for **GPS22.INP** file with multiple reference satellites

Table 2 – **GPS22.INP** field descriptions

Field	ID	Format	Description
A	LQUIT	(I5)	This is a flag that was used to indicate whether nav22 should be run or not. It should always be 0.
B	FID	(A4)	This is the database identifier. It is used to generate file names for the database files.
C	LSOLN	(I5)	This is a flag indicating what type of processing is to be performed. 0 = A Priori Mode, 1 = Solution Mode.
D	ICORR	(I5)	This is a flag indicating whether or not correlations are to be used.
E thru H		(3I5,F5.1)	This is the start time, fields are: Day of Year, Hour, Minute, Second.
I thru L		(3I5,F5.1)	This is the stop time, fields are: Day of Year, Hour, Minute, Second.
M	LFRQ	(I5)	flag indicating frequency to use. 1=L1, 2=L2, 3=L3, 4=L1-L2.
N	LTROP	(I5)	flag indicating whether or not to apply the troposphere correction. 0=NO, 1=YES.
O	LION	(I5)	flag indicating whether or not to model the ionosphere correction. 0=NO, 1=YES.
P	NJOUT	(I3)	Number of satellites to be omitted from solution.
Q	JOUTS	(*I3)	PRNs for each omitted satellite, as specified by field P,NJOUT. These are optional fields, and should only appear if NJOUT is not 0.
R	LSADJ	(I5)	Flag indicating whether satellite arc elements should be adjusted, 1=YES, 2=NO. If this flag is 1 then 6 more flags are read (6I5) indicating which elements should adjusted. These extra flags have not been assigned field numbers and do not appear in the example or description. These are optional fields, and should not appear unless LSADJ is not 0.
S	JREFS	(I5)	PRN of reference satellite.
T	NSTA	(I5)	Number of stations.
U	NAME	(A12)	Station Name
V	LSTA	(I5)	Station status, -1=OMIT, 0=REF, 1=SLV, 2=FID.
W	LCLK	(I5)	Clock flag, -1=MOD, 0=RNG, 1=SLV.
X	LHGTS	(I5)	Height flag, 0=FIX, 1=SLV.

Notes:

Fields A-R and T-X are the same as for a single reference. Field S, the single reference satellite, is set to 0. A line containing fields 1-5 is included for each of the multiple references.

Fields 2 thru 5 specify the start time for the reference satellite specified by field 1. Field 6 is added to indicate the end of the list, its value is always 0.

Input file for nav22: NAV22.INP

```
     0
b216
     1     1
   216    17     5    .0   216    13     5    .0
     4     2     1     0     0     1
     2
     2
aaf3            0
flt1            1
```

Fig. 10 – Example **NAV22.INP** file with single reference satellite

```
AAAAA
BBBB
CCCCCDDDDD
EEEEEFFFFFGGGGGHHH.HIIIIIJJJJJKKKKKLLL.L
MMMMMNNNNNOOOOOPPPPPQQQQQRRRRR
SSSSS
TTTTT
UUUUUUUUUUUUUVVVVV
UUUUUUUUUUUUUVVVVV
...
```

Fig. 11 – Fields for **NAV22.INP** file with single reference satellite

Table 3 – **NAV22.INP** field descriptions

Field	ID	Format	Description
A	LQUIT	(I5)	flag that was used to indicate whether nav22 should be run or not. It should always be 0.
B	FID	(A4)	database identifier. It is used to generate file names for the database files.
C	LCYC	(I5)	flag indicating what type of processing is to be performed. 1 = A Priori Mode, 2 = Solution Mode, 3 = Preprocess Mode.
D	LPLOT	(I5)	flag indicating what plots should be produced.
E thru H		(3I5,F5.1)	start time, fields are: Day of Year, Hour, Minute, Second.
I thru L		(3I5,F5.1)	stop time, fields are: Day of Year, Hour, Minute, Second.
M	LFRQ	(I5)	flag indicating frequency to use. 1=L1, 2=L2, 3=L3, 4=L3AVE.
N	LRNG	(I5)	flag indicating whether or not to do a range solution, and if so what type to do. 0=NONE, 1=POINT, 2=DIFF.
O	LTROP	(I5)	flag indicating whether or not to apply the troposphere correction. 0=NO, 1=YES.
P	LSWAP	(I5)	flag indicating whether or not an antenna swap was performed on the data.
Q	LBIAS	(I5)	flag indicating how to set phase bias. 0=Float, 1=Fix.
R	LFIX	(I5)	flag indicating if edits should be generated using doppler data.
S	JREF	(I5)	Reference satellite PRN.
T	NSTA	(I5)	Number of stations.
U	NAME	(A12)	Station Name.
V	ISET	(I5)	Station Status. -1=OMIT, 0=REF, 1=SLV.

Notes

Fields U and V are repeated for each station as specified by field T, NSTA.

```
     0
c208
     1    1
   208   20   27 30.0  208   22   34 00.0
     4    2    1    0    0    1
    00
    24  208   22   41   50
    16  209   00   55   00
    26  209   03   22   00
    21  999   00   00   30
     2
mag3            1
tul3            0
```

Fig. 12 – Example **NAV22.INP** file with multiple reference satellites

```
AAAAA
BBBB
CCCCCDDDDD
EEEEEFFFFFGGGGGHHH.HIIIIIJJJJJKKKKKLLL.L
MMMMMNNNNNOOOOOPPPPPQQQQQRRRRR
SSSSS
1111122222333334444455555
...
TTTTT
UUUUUUUUUUUUUVVVVV
UUUUUUUUUUUUUVVVVV
...
```

Fig. 13 – Fields for **NAV22.INP** file with multiple reference satellites

Fields A-R and T-V are the same as for a single reference. Field S, the single reference satellite, is set to 0. A line containing fields 1-5 is included for each of the multiple references. Fields 2 thru 5 specify the end time for the reference satellite specified by field 1.

Table 4 – **NAV22.INP** reference satellite list field description

Field	ID	Format	Description
1	IREF	(I5)	Reference Satellite.
2-4		(4I5)	Day of Year, Hour, Minute, Second.

Field 2, Day of Year, is set to 999 to indicate the last reference (see example).

ACKNOWLEDGMENT

XOMNI was developed under funding from the Oceanographer of the Navy and the technical direction of Mr. Kenneth Ferer.

REFERENCES

Aoki S., Guinot B., Kaplan G.H., Kinoshita H., McCarthy D.D, And Seidelmann P.K. (1982). "The New Definition Of Universal Time," Astronomy and Astrophysics, 105, 359-361.

Bate, R.R., Mueller, D.D. and White, J.E. (1971). Fundamentals Of Astrodynamics, Dover Publications, Inc., New York, NY.

Blewitt, G. (1990). An Automated Editing Algorithm for GPS Data, Geophys. Res. Lett., 17, 199-202.

Brozena, J.M., Mader, G.L. and Peters, M.F. (1989). Interferometric Global Positioning System: Three-Dimensional Positioning Source for Airborne Gravimetry, J. Geophys. Res., 94, 12153-12162.

Cartwright and Edden (1973). Geophys. J. Roy. Astr. Soc., 1973, 33, 253-264.

Cartwright and Tayler (1971). Geophys. J. Roy. Astr. Soc., 1971, 23, 45-74.

Chin, M. (1989). Argo Users Guide, Third Release, National Geodetic Survey.

Columbo, O.L. (1992). Airborne gravimetry, altimetry and GPS navigation errors, Proc. I.A.G. Symposium G3: Determination of the Gravity Field from Space and Airborne Methods, Vienna, Aug., 1991, Springer Verlag, NY, 1992.

Columbo, O.L. and Peters, M.F. (1992). Precision long-range DGPS for airborne surveys, GPS World, April 1992, 44-50.

DMA Technical Report TR8350.2 (1987). World Geodetic System.

Eissfeller, B. and Spietz, P. (1989). Basic Filter Concepts for the Integration of GPS and an Inertial Ring Laser Gyro Strapdown System, Manusc. Geod., 14, 166-182.

Escobal, P.R. (1965) Methods Of Orbit Determination, John Wiley & Sons Inc.

Faddeeva, V.N. (1959). Computational Methods Of Linear Algebra, Dover Publications, Inc.

Goad, C. (1985). Precise relative position determination using Global Positioning System carrier phase measurements in a non-difference mode, Proc. of the First Int. Symposium on Precise Positioning with Global Positioning System, NOAA, Rockville, MD.

Gurtner, W. and Mader, G.L. (1990). "The RINEX Format: Current Status, Future Developments," Proceedings of the Second International Symposium on Precise Positioning with the Global Positioning System.

Hein, G.W., Baustert, G., Eissfeller, B. and Landau, H. (1989). High-precision kinematic GPS differential positioning and integration of GPS with a ring laser strapdown inertial system, Navigation, 36, 77-98.

Hofmann-Wellenhof, B., Lichtenegger, H. and Collins, J. (1993). GPS Theory and Practice, 2nd Ed., Springer-Verlag, NY, 82-84.

Kaula, W.M. (1966). Theory Of Satellite Geodesy, , Blaisdell Publishing Co., Waltham, Massachusetts, 14-16.

Klobuchar, J.A. (1986). Design and characteristics of the GPS ionospheric time delay algorithm for single frequency users, Proc. Position, Location and Navigations Symp, IEEE, New York, New York.

Landau H. (1988). Zur Nutzung des Global Positioning Systems in Geodasie und Geodynamik: Modelbildung, Software-Entwicklung and Analyse, Ph. D. Thesis FAF University of Munich.

Leick, A. (1990). GPS satellite surveying, Pub. by John Wiley and Sons, New York, New York, 352 pp.

Mader, G.L. (1986). Dynamic positioning using GPS carrier-phase measurements, Manuscr. Geod., 11, 272-277.

Mader, G.L. (1990). Ambiguity function techniques for GPS phase initialization and kinematic solutions (abstract), Eos Trans. of Am. Geophys. Un., 71, 484.

Mader, G.L., Crump, D., Brozena, J.M. and Peters, M.F. (1991). Validation of Long-Range Kinematic GPS Positioning (abstract), Eos Trans. AGU, Spring Mtg. , 92.

Mader, G.L., Schenewerk, M.S. and Chin, M.M. (1990). OMNI 1.00 user's guide, National Geodetic Survey Report, National Oceanic and Atmospheric Administration, Rockville, MD USA.

Mueller, I.I. (1977). Spherical & Practical Astronomy As Applied To Geodesy, Ungar Publishing, 2nd Printing, 46 pp.

Rapp, R.H. (1984). Geometric Geodesy Part I, March 1984, Department Of Geodetic Science And Surveying, Ohio State University, 121 pp.

Robertson, D.S. (1975). Geodetic and Astrometric Measurements with VLBI, Doctoral Thesis, M.I.T, Available as a NOAA reprint, 49 pp.

Schwarz, K.P., Sideris, M.G., and Forsberg, R. (1987). Orthometric heights without leveling, J. Surveying Eng., 113, 28-40.

Soler, T. and Hothem, L.D. (1988). "Ordinate Systems Used In Geodesy: Basic Definitions And Concepts," Journal of Surveying Engineering, Vol.114, No.2, May 1988, 92 pp.

Van Dierendonck A.J., Russell, S.S., Kopitzke, E.R. and Birnbaum, M. (1980). "The GPS Navigation Message," Navigation GPS Special Issue, Vol. 1, The Institute of Navigation.